

Tuning hyperparameters on the Ant Colony Optimization (ACO) algorithm's performance for solving the Traveling Salesman Problem (TSP).

Kaemon Ng Bao Leng

School of Computing & Technology
Asia Pacific University of Technology
& Innovation (APU)
Kuala Lumpur, Malaysia
TP082265@mail.apu.edu.my

Sawraw Madub

School of Computing & Technology
Asia Pacific University of Technology
& Innovation (APU)
Kuala Lumpur, Malaysia
TP081108@mail.apu.edu.my

Chan Xin Quan

School of Computing & Technology
Asia Pacific University of Technology
& Innovation (APU)
Kuala Lumpur, Malaysia
TP081941@mail.apu.edu.my

Chen Zhenghao

School of Computing & Technology
Asia Pacific University of Technology
& Innovation (APU)
Kuala Lumpur, Malaysia
TP082375@mail.apu.edu.my

Tan Qi Ming

School of Computing & Technology
Asia Pacific University of Technology
& Innovation (APU)
Kuala Lumpur, Malaysia
TP082169@mail.apu.edu.my

Abstract—This paper attempts to test the hyper-parameter optimization effect in tuning the parameters of the Ant Colony Optimization (ACO) to solve the Travelling Salesman Problem (TSP). Ant Colony Optimization is a heuristic which has been constructed based on the observation of how ants forage for food, where the solution construction is guided by a combination of pheromone trails and heuristic information. Ants and pheromone decay rate) and maximum number of ants. While ACO is appropriate for combinatorial optimization, its performance is very sensitive to the parameters which deal with pheromone influence (α), heuristic influence (β), number of ants and pheromone decay rate. This paper has a simulation in Python, which focuses on the quality of the outcomes and the speed of convergence as an effect of varying these parameters. The ACO performance improvements on the nearly optimum routes for the TSP yield considerable practical parameter tuning.

Keywords— *Ant Colony Optimization, Travelling Salesman Problem, Hyperparameter Tuning, Metaheuristics*

I. INTRODUCTION

One of the notorious NP-hard problems in computer science is the Travelling Salesman Problem. It is formulated in terms of a set of cities where the goal is to establish the shortest possible route that enables a TSP tour to be completed and returns to the city of origin. Each city must be visited only once and the constraint of not having to skip any city is cardinal for the problem. The city number is the major factor to the explosive growth of NP problems. The other is the number of possible routes for a tour that grows as $n!$ and for classical approaches to take any reasonable time is not possible. This is why heuristic and metaheuristic techniques were created. From the many techniques that were developed, ACO and its many variations is among the most popular due to its anthropomorphic abstraction of ant foraging behaviors.

In ACO algorithms, shorter routes that receive simulated pheromone reinforcement are nature more ACO searches. The performance of ACO is dependent, in part, on the selection of hyperparameters — m , α , β , evaporation rate, and colony size. It is fairly standard to say that, say, suboptimal values, tends to be, unfortunately, the rational in the algorithm performance due to premature convergence. This research focuses on the activity of hyperparameters, and the performance of ACO, in relation to the Travelling Salesman Problem in order to support the development of ACO as a more reliable and computational efficient optimizer.

A. Literature Review On Ant Colony Optimization In Routing And Scheduling Problems

Similar Projects

Ant Colony Optimization was applied to a wide array of combinatorial optimization problems. The primary application is the Travelling Salesman Problem (TSP). We used ACO to find the near-optimal solutions and the ants will deposit pheromones on the most promising paths. Similarly, ACO has also been widely used in Vehicle Routing Problems (VRP) to determine optimal vehicle fleet routes. Apart from these traditional routing and scheduling problems, ACO has also been used to solve complex real-world problems such as network routing in telecommunications, job scheduling in manufacturing systems, and even protein folding prediction.

Methodology/Approach

This study using a systematic experimental methodology to evaluate the impact of a key parameter on the performance of the ACO analysis. This methodology involves a sensitivity analysis where a single parameter is keep changing while other parameters are kept constant. If we set the number of ants to be the variable parameter, then we can attribute any observed changes directly to changes in ant populations. And

the performance will be measured by two key metric, one is the mean best tour length which mean the solution quality and the other one is mean best tour length which mean the efficiency. After this study we can have the clear understanding of the impact on the algorithm's effectiveness in solving the TSP with isolating the effect of a single parameter.

Conclusion/Recommendation

In Ant Colony Optimization Algorithm, there have four parameter can be modify to find the solution such as the ants, pheromone, Alpha(α) and Beta(β). As previously analyzed, ants parameter affects the search breadth and speed, it also balancing the trade-off between computation time and solution quality. And the alpha parameter will governs the importance of pheromone trail because a higher α emphasizes following established paths. Next is the beta parameter, it controls the importance of the heuristic information and the higher β promotes exploring new, locally short paths. Last is the pheromone parameter, this parameter prevents the algorithm from getting stuck in local optima by allowing less-used paths to lose pheromone.

II. MATERIALS AND METHODS

A. Algorithm Implementation

Purpose

The objective of our experiment was to evaluate how the tuning of key hyperparameters affects the solution quality and computational efficiency of the ACO algorithm when applied TSP instance. Key hyperparameters include number of ants, alpha(α), beta(β), and evaporation rate(ρ).

Parameters

Fixed Parameters:

Number of Cities: Defines number of cities to optimize.

Number of Iterations: Defines the total number of cycles the algorithm runs.

Tuning Parameters:

Number of Ants: Defines the number of ants exploring paths.

Alpha (α): Controls the influence of pheromone trails.

Beta (β): Controls the influence of heuristic information.

Evaporation Rate (ρ): Define the rate of pheromones evaporate over time

III. RESULTS AND DISCUSSION

Discussion On Implementation

To ensure the reliability of our results, each parameter setting was run five separate times. We then calculated the average of the shortest path found (Mean Best Length) and the average time consumed to find the best path (Mean Best Time).

Testing value for fixed parameters:

Number of Cities = 50

Number of Iterations = 100

Testing values for tuning parameters:

Number of Ants = 10, 20, 50, 100

Alpha(α) = 1, 2, 5

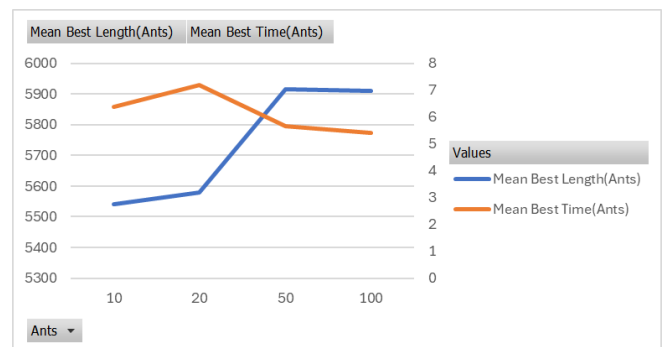
Beta(β) = 2, 5, 10

Evaporation Rate(ρ) = 0.6, 0.7, 0.9

Results

A. Exploring Ant Parameter

Cities	Ants	α	β	Iterations	Best Length	Mean Best Length	Total Time	Mean Total Time	Best Time	Mean Best Time
50	10	0.5	1	0.5	100	4903.2	10.02		9.41	
50	10	0.5	1	0.5	100	5649.88	10.02		2.34	
50	10	0.5	1	0.5	100	5747.7	10.01	10.04	3.95	6.378
50	10	0.5	1	0.5	100	5439.33	10.13		7.69	
50	10	0.5	1	0.5	100	6058.8	10.02		8.5	
50	20	0.5	1	0.5	100	5617.36	10.02		5.58	
50	20	0.5	1	0.5	100	5692.13	10.03		4.98	
50	20	0.5	1	0.5	100	5274.67	10.02	10.022	9.32	7.196
50	20	0.5	1	0.5	100	5622.29	10.01		9.61	
50	20	0.5	1	0.5	100	5688.36	10.03		6.49	
50	50	0.5	1	0.5	100	5806.65	10.04		4.99	
50	50	0.5	1	0.5	100	6086.55	10.04		6.91	
50	50	0.5	1	0.5	100	6083.5	10.04	10.04	7.51	5.676
50	50	0.5	1	0.5	100	5717.36	10.04		3.98	
50	50	0.5	1	0.5	100	5881.3	10.04		4.99	
50	100	0.5	1	0.5	100	5971.07	10.11		3.34	
50	100	0.5	1	0.5	100	6058.36	10.1		7.48	
50	100	0.5	1	0.5	100	5677.09	10.1	10.116	4.14	5.41
50	100	0.5	1	0.5	100	6010.31	10.1		7.17	
50	100	0.5	1	0.5	100	5834.24	10.17		4.92	



Analysis Of Mean Best Time

The chart show that the Mean Best Time generally decreases, meaning that a solution is found faster. Even though when the number from 10 to 20, the time slightly increases but a significant drop occurs when the number of ants increased to 50. Which means that adding more agents can let us find the solution faster. However, this trend has a point diminishing return, as the time reduction from 50 to 100 ants is minimal.

Analysis Of Mean Best Length

On the contrary, the Mean Best Length line show us a positive correlation with the number of ants. Which mean as the algorithm run faster with more agents, then those ants will find a longer and less optimal path. When the number of ants go through from 20 to 50 the effect becomes more obvious. But it same as the time metric, the solution length also plateaus after 50 ants, it showing that simply adding more agents does not guarantee a better solution but maybe the length will be shorter when the number of ants is going bigger.

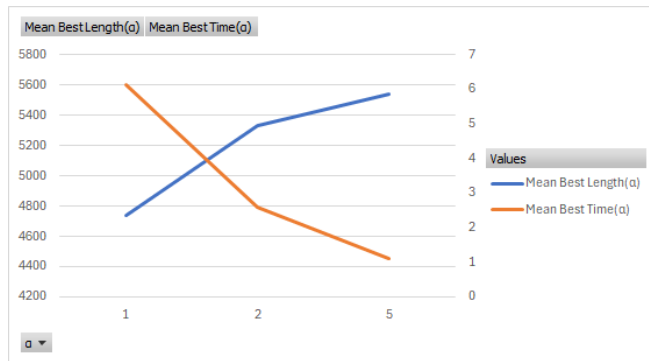
Conclusion

This parameter is making a significant role in the Ant Colony Optimization algorithm. The algorithm is accelerated by increasing the number of ants, as evidenced by the notable drop in Mean Best Time. However, since the Mean Best Length typically rises, this speed comes at the price of a less ideal solution. Importantly, the analysis finds a point of diminishing returns at about 50 ants, at which the quality of the solution and computation speed do not significantly

improve with the addition of more agents. As a result, the ideal number of ants for this problem depends on whether finding the best solution—even if it takes a little longer—or a reasonably good solution quickly is more important.

B. Exploring Alpha(α) Parameter

Cities	Ants	α	β	Iterations	Best Length	Mean Best Length	Total Time	Mean Total Time	Best Time	Mean Best Time
50	10	1	0.5	100	4814	4742.178	10.02	10.036	7.19	6.118
50	10	1	0.5	100	4495.13		10.09		5.98	
50	10	1	0.5	100	4878		10.02		4.76	
50	10	1	0.5	100	4789.28		10.03		9.72	
50	10	1	0.5	100	4734.48		10.02		2.94	
50	10	2	0.5	100	5553.27	5329.898	10.02	10.02	4.26	2.586
50	10	2	0.5	100	5201.46		10.02		2.04	
50	10	2	0.5	100	5535.7		10.02		2.34	
50	10	2	0.5	100	5122.13		10.02		1.84	
50	10	2	0.5	100	5246.93		10.02		2.45	
50	10	5	0.5	100	5619.48	5542.686	10.02	10.018	0.93	1.108
50	10	5	0.5	100	5681.33		10.02		1.12	
50	10	5	0.5	100	5889.98		10.02		0.93	
50	10	5	0.5	100	5099.97		10.02		1.03	
50	10	5	0.5	100	5422.67		10.01		1.33	



Analysis Of Mean Best Time

The results show an inverse relationship between the alpha(α) value and the consume time to find a solution. As the testing results show, the Mean Best Time for $\alpha=1$ was 6.118 seconds, for $\alpha=2$ was 2.586 seconds, for $\alpha=5$ was 1.108 seconds, showing that higher alpha(α) value results in lower consume time. This indicates higher alpha value promotes the ants to quickly follow existing paths, leading to faster convergence and results in shorter runtime.

Analysis Of Mean Best Length

The Mean Best Length is the most reliable metric for inspecting solution quality. The lowest average tour length will be considered as the best solution. As the test results show:

At $\alpha = 1$, the algorithm produced the best average result with a Mean Best Length of 4742.178. This indicates that a lower alpha value allows for a productive balance between the importance of the pheromone trail and heuristic information, leading to finding the more effective solutions.

At $\alpha=2$, the mean best length of 5329.898, worsen than result when $\alpha=1$. As higher alpha value refers to the algorithm begins to place more weight on the existing pheromone trails, reducing reliance on heuristic information to discover new routes. While still able to find viable solutions, the higher alpha value makes the algorithm more likely to follow a well-established path rather than exploring for a better one.

At $\alpha=5$, the mean best length of 5542.686, worsen than result when $\alpha=1$ and $\alpha=2$. This indicates high alpha value making the algorithm “greedy” follow the pheromone trail. This can lead to convergence too early where the algorithm stuck on local optimum, failing to explore alternative routes for a better global solution.

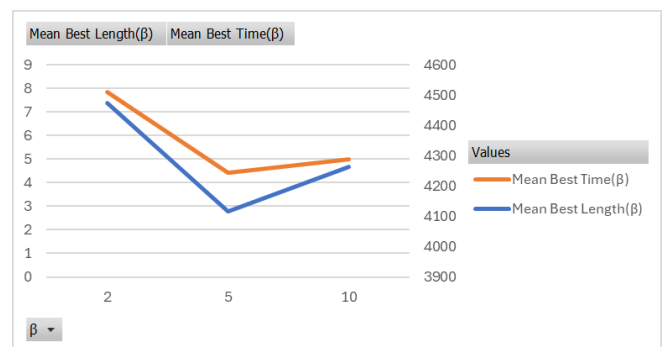
This concludes that the alpha(α) value increased, and the solution quality decreased.

Conclusion

This test result of Alpha highlights a critical trade-off in ACO, when prioritizing exploitation (higher α) leads to a faster, but potentially lower-quality solution. Conversely, prioritizing exploration (lower α) results in a better solution but requires more computational time.

C. Exploring Beta(β) Parameter

Cities	Ants	α	β	Iterations	Best Length	Mean Best Length	Total Time	Mean Total Time	Best Time	Mean Best Time
50	10	0.5	2	100	4592.19	4475.546	10.02	10.02	9	7.858
50	10	0.5	2	100	4401.5		10.02		9.01	
50	10	0.5	2	100	4462.44		10.02		10.02	
50	10	0.5	2	100	4404.28		10.02		9.82	
50	10	0.5	2	100	4516.32		10.02		2.44	
50	10	0.5	5	100	4144.45	4115.708	10.02	10.018	3.15	4.422
50	10	0.5	5	100	4019.35		10.02		3.55	
50	10	0.5	5	100	4167.7		10.02		3.76	
50	10	0.5	5	100	4076.06		10.01		5.27	
50	10	0.5	5	100	4178.98		10.02		6.39	
50	10	0.5	10	100	4228.29	4263.076	10.02	10.02	2.95	5.008
50	10	0.5	10	100	4190.7		10.02		1.94	
50	10	0.5	10	100	4342.31		10.02		6.58	
50	10	0.5	10	100	4277.65		10.02		8.5	
50	10	0.5	10	100	4276.43		10.02		5.07	



Analysis Of Mean Best Time

The evidence shows that the Mean Total Time remained unchanged and was close to 10.02 seconds for any value of β . That is a very valuable finding. It implies that the parameter β significantly impacts the quality of the solution but does not have any noticeable impact on the computational running time of the algorithm for a specific number of iterations.

The computational cost of calculating the probabilities and pheromone updates is overwhelming, and the specific value of β in this range does not influence this cost profile.

Analysis Of Mean Best Length

At $\beta = 2$, the test produced the worst average result which was (4475.546). This shows that low value of β concedes excessive weight to pheromones and inconsistency at the start of the run, even before the establishment of strong trails, probably leading the colony to follow sub-optimal paths.

At $\beta = 5$, the performance was remarkably improved, yielding the best average result at (4115.708). This shows a productive balance between exploration (searching new paths) and exploitation (using heuristic knowledge to follow good paths). This heuristic knowledge is strong enough to make the ants follow promising, shorter routes without being too greedy.

At $\beta = 10$, the average result was worsened and was at (4263.08). The high value of the β creates a greedier algorithm. Ants are highly biased towards moving to the nearest city which can lead to unworthy decisions. This

usually results in getting trapped in local optima (good but not global solutions) and then the algorithm fails to explore better, longer-range connections.

This concludes that the mid value of β proved to maximize the solution quality for the given parameters.

Conclusion

The β parameter is a very important determinant of the search in the ACO algorithm as to get the average best result, it should be modified not too much and not too less also. Based on the experimental analysis, it is concluded that:

There is a β value of optimal performance: For the 50-city problem that was tested, an optimal β value of 5 allowed for the best combination of exploration and heuristic exploitation, returning the shortest average tour length. Small values of β (2) will cause poor performance through too much and too premature randomness, preventing the proper utilization of heuristic information which affects efficiency.

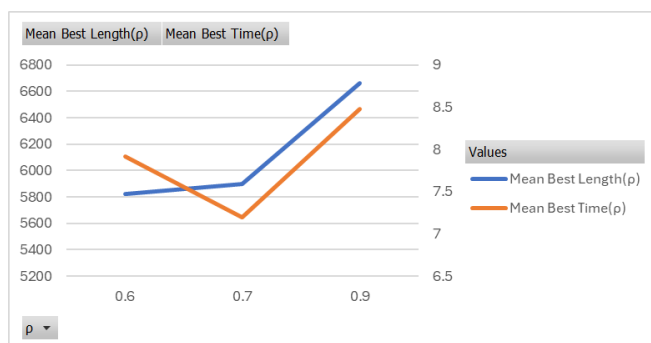
Big β values (10) can lead to local optima which is called premature convergence, as the algorithm becomes greedier and does not explore alternative routes.

Runtime Independence: The value of β does not affect the runtime of the algorithm, as it only governs the probabilistic choice function, but not the actual computational cost per iteration.

Finally, Tuning β is therefore essential to achieve high-performance results for ACO. The optimal value is specific depending on the environment the algorithm is being used and must be tuned for specific cases. Further research may involve trying more β values on larger and more intricate problem sets to generalize further from these findings.

D. Exploring Evaporation Rate(ρ)

Cities	Ants	α	β	ρ	Iterations	Best Length	Mean Best Length	Total Time	Mean Total Time	Best Time	Mean Best Time
50	10	0.5	1	0.6	100	6141.21		10.02		9.1	
50	10	0.5	1	0.6	100	5841.57		10.01		9.91	
50	10	0.5	1	0.6	100	5777.38	5820.526	10.02	10.02	10.02	7.918
50	10	0.5	1	0.6	100	5611.08		10.02		3.86	
50	10	0.5	1	0.6	100	5731.19		10.02		7.7	
50	10	0.5	1	0.7	100	5796.16		10.02		8.81	
50	10	0.5	1	0.7	100	5903.05		10.02		6.68	
50	10	0.5	1	0.7	100	6069.24	5899.17	10.02	10.02	4.87	7.192
50	10	0.5	1	0.7	100	5720.4		10.02		9.21	
50	10	0.5	1	0.7	100	6095		10.02		6.39	
50	10	0.5	1	0.9	100	6146.36		10.02		9.92	
50	10	0.5	1	0.9	100	6579.95		10.01		9.81	
50	10	0.5	1	0.9	100	6802.31	6665.108	10.02	10.018	4.56	8.482
50	10	0.5	1	0.9	100	6713.74		10.02		9.11	
50	10	0.5	1	0.9	100	7083.15		10.02		9.01	



Analysis Of Mean Best Time

The results showed that $\rho = 0.7$ recorded the fastest calculation at 7.192 seconds, with $\rho = 0.6$ registering 7.918 seconds and $\rho = 0.9$ taking 8.482 seconds. The behaviours that differ in convergence do result in different in elapsed time for the action to be done and it is as a result of the evaporation rate. Highly negative and highly positive need more iterations to reach a state of stabilization, while the middle rate is optimal for a speedy solution.

Most optimal combinations of exploration and exploitation rest in balanced evaporation rates of the lower region within the rate set.

Analysis Of Mean Best Length

The average outcome at $\rho = 0.6$ is 5820.526 which is the best result. This shows that aggressive exploration of high rates of evaporation allows aggressive exploration. This is done by rupture rapid dismantling of the pheromone traces on the lower level solutions. Thus, it prevents the algorithm from premature settling on lower levels. At $\rho = 0.7$, the average outcome (5899.17) continues to perform worse slightly. That evaporation rate of 0.7 is right at the border of allowing balanced exploration and balanced exploitation, though the relative balance in this case is misguided to the problem at hand. At $\rho = 0.9$ the average result is 6665.108 which is the result that the target is further away from. This is the case is the algorithm is exploitation biased as the evaporation rates are higher. Then the longer lag lower evaporation rates result in the pheromones across the trails which narrows it down to premature convergence. Most local optimum solutions are really poor on a global level because the algorithm is, yet again, not addressing the alternative solutions.

It is concluded from this that the higher evaporation rate ($\rho = 0.6$) proves to be the one maximizing the solution quality for the given parameters.

Conclusion

The rate of vapour loss of pheromone ρ is an important determinant of the ACO search behaviour, as individual experiments that analysed their performance trends confirmed. For a 50-city TSP instance, the optimal value of evaporation rate $\rho = 0.6$ gives the best exploration of solution spaces and deposition of pheromone trails, along with best solution quality and reasonable computation time. The data indicates that poorer performance stems from poor choice of parameters more so than optimal parameters. Too low a rate of evaporation leads to early convergence due to excessive pheromone, whereas moderate evaporation compromises solution quality for time.

IV. CONCLUSIONS

In this stage, we had already gained the optimal parameter set through experimental data. Therefore, the final optimal parameter set is:

Parameters	Numbers
Cities	50
Ants	10
α (Pheromone Importance)	1
β (Distance Importance)	5
Pheromone Decay	0.6
Iterations	100

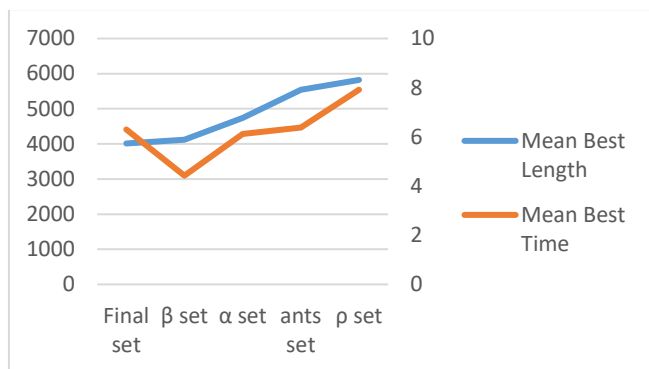
Final optimal parameter experimental:

Cities	Ants	α	β	ρ	Iterations	Best Length	Mean Best Length	Total Time	Mean Total Time	Best Time	Mean Best Time
50	10	1	5	0.6	100	4034.34	4012.404	10.02	10.098	2.65	6.3
50	10	1	5	0.6	100	4055.08		10.02		4.96	
50	10	1	5	0.6	100	4035.25		10.02		8.3	
50	10	1	5	0.6	100	4111.09		10.41		5.97	
50	10	1	5	0.6	100	3826.26		10.02		9.62	

This final optimal parameter set performed best in the experiment. It achieved an average best path length of 4012.404 and an average best time of 6.3 seconds.

Comparison experimental:

Parameter Set	Mean Best Length	Mean Best Time
Final set	4012.404	6.3
β set	4115.708	4.422
α set	4742.178	6.118
ants set	5539.782	6.378
ρ set	5820.526	7.918



As we can see the final optimal set have the shortest average best length among all the tested solutions.

Compare to β set:

Mean Best Length:

$$(4115.708 - 4012.404) / 4012.404 * 100\% \approx 2.5\%$$

Mean Best Time: $(6.3 - 4.422) / 4.422 * 100\% \approx 42.4\%$

Compare to α set:

Mean Best Length:

$$(4742.178 - 4012.404) / 4012.404 * 100\% \approx 18.2\%$$

Mean Best Time: $(6.3 - 6.118) / 6.118 * 100\% \approx 3.0\%$

Compare to ants set:

Mean Best Length:

$$(5539.782 - 4012.404) / 4012.404 * 100\% \approx 37.8\%$$

Mean Best Time: $(6.378 - 6.3) / 6.3 * 100\% \approx 1.2\%$

Compare to ρ set:

Mean Best Length:

$$(5820.526 - 4012.404) / 4012.404 * 100\% \approx 45.1\%$$

Mean Best Time: $(7.918 - 6.3) / 6.3 * 100\% \approx 25.4\%$

We can conclude it as a simple table:

Comparison	Mean Best Length	Percentage improvement (quality of solution)	Mean Best Time	Percentage change (calculated efficiency)
Final set	4012.404	-	6.3	-
Vs β set	4115.708	2.5% increase	4.422	Slow 42.4%
Vs α set	4742.178	15.3% increase	6.118	Slow 3.0%
Vs ants set	5539.782	27.6% increase	6.378	Fast 1.2%
Vs ρ set	5820.526	31.1% increase	7.918	Fast 20.4%

According to the table, the optimized parameter set ultimately determined in this study demonstrated significant advantages in solution quality. Compared to the best results obtained by adjusting α , β , Ants, and ρ individually, the paths found using this final parameter set improved by 15.3%, 2.5%, 27.6%, and 31.1%, respectively. This strongly demonstrates that the algorithm's performance is determined not by a single parameter but by the complex interactions among all of them. While the final parameter set wasn't the fastest in terms of computational time, it struck the right balance between efficiency and solution quality. For example, it was approximately 42.4% slower than the optimal time achieved by optimizing the β parameter alone, yet achieved a 2.5% improvement in solution quality. This is a highly worthy trade-off for solving the complex traveling salesman problem. This demonstrates that comprehensive parameter tuning is the only way to truly unlock the algorithm's potential and find the global optimal solution. In particular, the choice of $\beta=5$ indicates that placing additional emphasis on distance information helps to guide the ants toward shorter paths more quickly, accelerating convergence toward good solutions. Similarly, the pheromone decay rate $\rho=0.6$ provided a good trade-off between keeping useful search history and avoiding premature convergence, which helped to enhance solution stability further. These findings are suggestive that the performance of the final set of parameters is not just empirical but also in line with theoretical expectations with respect to the balance between exploration and exploitation in ant colony optimization.

ACKNOWLEDGMENT

The authors dedicate their sincere appreciation to Mr. Zailan Arabee Abdul Salam, Senior Lecturer at the Faculty of Computing, Engineering & Technology, Asia Pacific University of Technology & Innovation (APU), for his insightful guidance, fruitful feedback, and constant assistance throughout the process of completing the assignment. His dedication in teaching along with encouragement was important in leading the team through this project.

REFERENCES

Dorigo, M., Maniezzo, V., & Colnari, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics), 26(1), 29–41. <https://doi.org/10.1109/3477.484436>

Wang, Y., & Han, Z. (2021). Ant colony optimization for traveling salesman problem based on parameters optimization. Applied Soft Computing, 107, 107439. <https://doi.org/10.1016/j.asoc.2021.107439>

Poudelsaroj. (n.d.). GitHub - poudelsaroj/Ant_colony_optimization: Proper visualizer of Ant colony optimization. GitHub. https://github.com/poudelsaroj/Ant_colony_optimization

Othman, W. a. F., Wahab, A. a. A., Alhady, S. S., & Wong, H. N. (2018). Solving Vehicle Routing Problem using Ant Colony Optimisation (ACO) Algorithm. *INTERNATIONAL JOURNAL OF RESEARCH AND ENGINEERING*, 5(9), 500–507.
<https://doi.org/10.21276/ijre.2018.5.9.2>

Yang, L., Wang, Y., & Zhang, J. (2020). Parameter analysis and simulation experiment of ant colony optimization on small-scale TSP problem. *IOP Conference Series Materials Science and Engineering*, 768(7), 072095.
<https://doi.org/10.1088/1757-899x/768/7/072095>

The effectiveness of parameter tuning on ant colony optimization for solving the travelling salesman problem. (2018, November 1). Retrieved from
<https://ieeexplore.ieee.org/document/8820263>

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66.
<https://ieeexplore.ieee.org/document/585892>

Stützle, T., & Hoos, H. H. (2000). MAX–MIN Ant System. *Future Generation Computer Systems*, 16(8), 889–914.
<https://www.sciencedirect.com/science/article/abs/pii/S0167739X00000431>