Improvements on the Ant-System: Introducing the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System

Thomas Stützle* and Holger Hoos
TH Darmstadt, FB Informatik, FG Intellektik
Alexanderstr. 10, D-64283 Darmstadt, Germany
{tom,hoos}@informatik.th-darmstadt.de

Abstract

In this paper we present $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) that improves on Ant System. $\mathcal{MM}$AS is a general purpose heuristic algorithm based on a cooperative search paradigm that is applicable to the solution of combinatorial optimization problems. In the experiments we apply $\mathcal{MM}$AS to symmetric and asymmetric Traveling Salesman Problems. We describe in detail the improvements on Ant System, discuss the addition of local search to $\mathcal{MM}$AS, and report on our computational results, showing that our system also improves over other variations of Ant System.

1 Introduction

Ant System (AS), introduced originally in [1, 2], is a cooperative search algorithm inspired by the behaviour of real ants, that are able to find good solutions to shortest path problems between food sources and their home colony. Ants usually communicate via pheromones, i.e. aromatic substances, that they may lay down in some quantity. An ant's tendency to choose a specific path is positively correlated to the intensity of a found trail. The pheromone trail evaporates over time if no more pheromone is laid down by other ants. If many ants choose a certain path and lay down pheromones, the intensity of the trails increases attracting more ants. The behaviour of ant colonies is imitated to some extent by AS by using simple agents, called *ants*, that may communicate among themselves via a mechanism inspired by the pheromone trails. This allows the application of this search metaphor to the solution of combinatorial optimization problems [3]. Here we present $\mathcal{MM}$AS, an improvement over Ant System at hand of the application to symmetric and asymmetric Traveling Salesman Problems (TSPs) contained in TSPLIB (`ftp://ftp.iwr.uni-heidelberg.de/pub/tsplib`).

* Corresponding author

The paper is organized as follows. We introduce the TSP in Section 2 and explain how AS can be used to solve TSPs. In Section 3, we introduce $\mathcal{MM}$AS and discuss the influence of some parameters on its performance. In Section 4 we discuss the addition of local search heuristics to $\mathcal{MM}$AS and conclude in Section 5.

2 The Ant System for TSPs

A TSP can be represented by a complete weighted directed graph $G = (\mathcal{V}, \mathcal{A}, d)$ where $\mathcal{V} = \{1, 2, \ldots, n\}$ is a set of nodes (cities), $\mathcal{A} = \{(i,j) || (i,j) \in \mathcal{V} \times \mathcal{V}\}$ a set of arcs, and $d : \mathcal{A} \mapsto \mathbb{N}$ a weight function, associating a positive integer weight d_{ij} with every arc (i,j) which can be interpreted as the distance between i and j. The aim is to find a shortest Hamiltonian Cycle, i.e., a cyclic route of minimal length visiting every node exactly once. For symmetric TSPs, the distances between nodes are all independent of the direction, i.e. for every pair of nodes we have $d_{ij} = d_{ji}$. If this condition does not hold, we have the more general case of an asymmetric TSP (ATSP). TSP is a $\mathcal{NP}$-hard optimization problem which has many applications and is extensively studied in the literature [8]. It also frequently serves to test new algorithmic ideas, one more reason to choose it here as an example application.

To solve TSPs, AS uses pheromone trails τ_{ij} associated with each arc (i,j). Initially, each of the m ants is set on some randomly selected city and starts constructing a tour from there.[1] A tour is successively built by choosing the next node probabilistically according to a probability distribution[2] proportional to:

$$p_{ij} \sim \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta} \ \text{ if } j \text{ not yet visited, else } 0 \qquad (1)$$

where η_{ij} is a local heuristic function which is defined as $\eta_{ij} = 1/d_{ij}$ in AS. The parameters α and β determine the relative influence of the

[1] For a more detailed description of the Ant System algorithm the reader is referred to [3].
[2] induced by normalization

trail strength and the heuristic information. To keep track of the cities already visited, every ant maintains a tabu list, in which the partial tour is stored. The trails are updated after all ants have constructed a complete tour and have calculated the corresponding tour length L_k. Every ant is allowed to lay down a constant quantity Q of pheromone and the trail intensities are updated according to

$$\tau_{ij}^{new} = \rho \cdot \tau_{ij}^{old} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (2)$$

where ρ is the persistence of the trail, thus $(1-\rho)$ simulates the evaporation. The amount $\Delta\tau_{ij}^{k}$ is equal to Q/L_k if arc (i,j) is used by ant k in its tour, otherwise zero. Frequently used arcs and arcs contained in short tours receive a high amount of pheromone. This can be interpreted as a learning of *good* arcs where arcs leading to short tours receive high amounts of trail and may be selected more often. The two basic steps *tour construction* according to (1) and *trail update* according to (2) are then simply repeated for a given number of iterations.[3]

The most important part in AS is the treatment of the trail intensities. If only few trail intensities are very high, the ants are very likely to choose the next arc among those with a very high trail intensity. In practice, the long term effect of the trail intensities is to reduce the size of the effective search space by concentrating the search on a relatively small number of arcs. To characterize the amount of exploration AS still performs, the mean λ-branching factor can be used, for details see [7]. If the mean λ-branching factor is very low, practically no more new tours are build, thus leading to a stagnation of the search.

3 $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System

$\mathcal{MM}$AS was inspired by some of our observations while experimenting with AS. With AS, a modified strategy in which only ants with very good tours are allowed to update the trails, showed promising results. Yet with this variation the major problem was premature stagnation of the search, leading to suboptimal tours. We conjectured that one of the most promising possibilities to improve AS should be to exert more influence directly onto the trail strengths. Thus, we introduced explicit maximum and minimum trail strengths on the arcs, hence the name $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. The maximum and minimum trail limits are chosen in a problem-dependent way. For the maximum trail limit $\tau_{\max}$ we use $\tau_{\max}(t) = n/L_{\min}$, where $L_{\min}$ is the minimal tour length found during the run of the algorithm. The minimum trail limit $\tau_{\min}$ is fixed to $\mathtt{const}/\ \emptyset \cdot n^2$, where $\emptyset$ is the average arc length.

As we use as a lower limit $\tau_{\min}$ the probability that a specific arc is chosen may get very small, but still will never be equal to zero. This leads to a higher degree of exploration. $\mathcal{MM}$AS differs also in one more important issue from AS. We only allow the best ant in each iteration to update the trail intensity according to formula 2. The trail strengths in $\mathcal{MM}$AS are initialized to $\tau_{\max}$ for all arcs. After each iteration the evaporation will reduce the trail strength by a factor ρ and only the trails on arcs of the best tours are allowed to increase their intensities or maintain them at the upper level. Hence, arcs that do not receive any or very rare reinforcement will continuously lower their trail strength and be selected more rarely by the ants.

3.1 Experiments with $\mathcal{MM}$AS

A very important parameter to influence the performance of $\mathcal{MM}$AS is ρ which determines how fast the trails strength on the arcs may decrease. We present the results obtained with different values for ρ and different number of iterations in some more detail in Table 1.[4] The results are presented at hand the problem instance `ry48p` as it was small enough to run many experiments.[5] The values for ρ should be rather high, as usually by increasing ρ better average performance is obtained. The particular parameter values for ρ also depend on the number of iterations allowed, as can be seen by comparing the performance for $\rho = 0.99$ with 1,000 and 2,500 iterations. A careful analysis shows that the average of the branching factor at which the best solution is found in a run is rather low. For the runs with $\rho = 0.99$ and 1,000 iterations the mean branching factor is still too high and thus by allowing more iterations (here 2,500) better results can be obtained. By this example we also can see that the branching factor is an essential descriptive parameter to judge the performance of $\mathcal{MM}$AS and to see whether still improvements can be expected. For higher values for ρ, like $\rho = 0.995$, and more iterations, better average results may be obtained at a rather low standard deviation. Yet, the disadvantage are the considerably higher run times. Note that the average solution quality of $\mathcal{MM}$AS with reasonable parameters for ρ is considerably better than the best results obtainable with AS.

Now we shortly indicate our results corresponding to other parameter choices for $\mathcal{MM}$AS. We

[3] We refer to the complete cycle of tour construction and trail update as one iteration

[4] If not indicated otherwise the parameter values are chosen as: $\alpha = \beta = 1$, $Q = 1$, and the number of ants m is chosen equal to the number of cities n.

[5] Note, that this instance was solved to optimality only recently in [5], the optimal tour length being 14,422.

Table 1: $\mathcal{MM}$AS on **ry48p.atsp**, 25 runs. **best** is the shortest tour found in 25 runs, **avg.best** is the average tour length and σ the standard deviation, **iterations** is the number of iterations performed.

ρ	best	avg.best	σ	iterations
0.95	14482	14860.76	230.60	1000
0.98	14422	14693.20	168.92	1000
0.99	14689	14938.68	187.67	1000
0.95	14422	14827.60	235.57	2500
0.98	14422	14666.16	171.03	2500
0.99	14495	14609.48	109.58	2500
0.995	14495	14546.60	22.01	25000

Table 2: Experimental results with 960.000 tour construction on **ry48p.atsp**. **quality** is the percentage deviation from the optimal solution. Ant Colony System 15 runs, others 25 runs.

Version	ρ	avg.best (quality)	σ
AS	0.3	14622.24 (1.38%)	126.03
ACS	–	14565.45 (0.99%)	115.23
$\mathcal{MM}$AS	0.995	14571.68 (1.03%)	90.72
$\mathcal{MM}$AS + sm	0.99	14461.40 (0.27%)	39.27

observed that after adding the maximum and minimum trail limits, it is significanlty better to allow only the best ant to update the trails. Good parameter values for β and α are $\alpha = 1.0$ and $1 \leq \beta \leq 5$. It is also interesting to note that the best value for β depends on the problem instance, whereas $\alpha = 1$ always seemed to work best. If not indicated otherwise, our results detailed below are obtained with $\alpha = 1.0$ and $\beta = 1.0$. We also investigated the influence of the values for $\tau_{\min}$ and $\tau_{\max}$. It turned out that the particular settings for $\tau_{\min}$ have less influence on the performance of $\mathcal{MM}$AS than the values for $\tau_{\max}$.

3.2 Smoothing of the Trails

As we have seen, the best tours usually are found at rather small values for the mean branching factor. Yet, if the branching factor is too low, only few new tours are built, thus leading to a very small exploration of possibly better tours. This lead us to the following idea: If $\mathcal{MM}$AS shows stagnation as indicated by the mean λ-branching factor we have to adjust the trail intensities in such a way that the exploration of new tours is increased again. We adjust the trails according to a *proportional update:* The trail intensity on arc (i, j) is increased proportionally to the difference between $\tau_{\max} - \tau_{ij}(t)$.

An advantage of the proportional update is that we do not completely forget the trails learned so far. Its overall effect is that by increasing the trail intensities the probability distribution for the selection of the next node is influenced in such a way that the exploration of new tours is higher. We call this approach *smoothing* of the trails as the differences between high and low trail intensities become less pronounced, i.e., smoother. A major advantage of smoothing is that it makes $\mathcal{MM}$AS more robust against premature convergence for a broader range of parameter values (especially ρ). In Table 2 we present the best average tour quality obtained with several extensions of AS, including Ant Colony System (ACS) [6], the so far best improvement over AS. It can be noted that $\mathcal{MM}$AS with additional trail smoothing ($\mathcal{MM}$AS+sm) performs best.[6]

Table 3: Results on contest problems, Ant Colony System, 15 runs

Problem	best	avg.best	σ
ry48p.atsp (48)	14422	14565.45	115.23
ft70.atsp (70)	38781	39099.05	170.32
kro124p.atsp (100)	36241	36857.00	521.19
ftv170.atsp (170)	2774	2826.47	33.84
eil51.tsp (51)	426	428.06	2.48
kroA100.tsp (100)	21282	21420	141.72
d198.tsp (198)	15888	16054	71.15

Table 4: Results on contest problems. 25 independent runs. 20,000 Iterations for ATSPs, 10,000 for symmetric TSPs

Problem	best (quality)	avg.best	σ
ry48p	14422 (0.0%)	14461.64	39.27
ft70	38690 (0.04%)	38903.44	149.85
kro124	36416 (0.50%)	36594.36	156.03
ftv170	2826 (1.74%)	2836.40	14.91
eil51	426 (0.0%)	427.2	1.13
kroA100	21282 (0.0%)	21352.05	50.30
d198	15960 (1.14%)	16065.95	73.82

In Table 4 we present the results obtained with $\mathcal{MM}$AS for some problems of the First International Contest on Evolutionary Optimization [4]. We only present results for the maximum number of tour constructions. The parameter settings are $\rho = 0.99$, $\alpha, \beta = 1.0$, and $m = n$ except for **ftv170.atsp**, where we chose $m = n/2$. In Table 3 we reproduced the results obtained with ACS in [6]. With execption of the problems **ftv170.atsp** and **d198.tsp**, the average performance of the $\mathcal{MM}$AS achieved over 25 independent runs is better and its standard deviations are generally smaller. The results show the very good performance of the $\mathcal{MM}$AS when compared to the best improvement so far over AS.

4 A local search $\mathcal{MM}$AS

We now investigate the performance of $\mathcal{MM}$AS when an additional a local search is performed. The reason for adding local search algorithms to $\mathcal{MM}$AS is twofold. On the one hand, we want to enhance the performance by adding local search yielding a faster convergence of the algorithm and

[6]For larger problem sizes the differences especially to AS become even larger.

Table 5: Results for symmetric TSPs and ATSP for $\mathcal{MM}$AS + local search. 1000 iterations, $m = 10$.

Problem	best	avg.best	σ
eil51	426 (0.0%)	426.3 (0.07%)	0.48
kroA100	21282 (0.0%)	21285.3 (0.015%)	7.60
d198	15807 (0.17%)	15824.80 (0.28%)	10.96
lin318	42762 (1.74%)	42892.5 (2.05%)	110.7
pcb442	51591 (1.60%)	52114.30 (2.63%)	239.1
p43	5620 (0.0%)	5620.60 (0.011%)	0.516
ry48p	14422 (0.0%)	14464.60 (0.30%)	43.75
ft70	38679 (0.016%)	38713.80 (0.11%)	22.45
kro124p	36235 (0.04%)	36387.40 (0.43%)	104.8

an earlier detection of high quality solutions. On the other hand, $\mathcal{MM}$AS should be able to construct good initial tours for the following local search phase, guiding the local search procedure towards better solutions. For symmetric TSP we implemented the so called 2-opt heuristic. For ATSPs we used *reduced 3-opt*, which is based on modified 3-opt exchanges that allow reinserting arcs without reversing the direction of a partial tour.

In our experiments we used 10 ants for each problem and, except for problems `lin318` and `pcb442`, all ants performed local search. In these experiments local search was performed in each iteration. In Table 4, results for symmetric TSPs with additional 2-opt and for ATSPs with reduced 3-opt are given. For the larger symmetric TSP `pcb442` (with 442 cities) only the best ant in each iteration is allowed to perform 2-opt and to reinforce the trail.

We also verified that it is better to use local search in addition with $\mathcal{MM}$AS than to use it in addition to AS. We also investigated the performance of greedy random tour construction followed by a local search phase, a simple method for the solution of combinatorial optimization problems.[7] For the problem `ry48p.atsp` we obtained as the best result 14,782 after 1,000,000 applications of the reduced 3-opt.

5 Discussion and Conclusion

In this article we presented $\mathcal{MM}$AS and showed that it performs much better than AS and at least at the same level of performance as ACS. By the addition of minimum and maximum trail limits, $\mathcal{MM}$AS offers a more direct control over the trails strength. Our approach of smoothing the trail led to an effective balance between exploration of new tours and exploitation of the learned trail strength. Additionally, we showed that the performance of $\mathcal{MM}$AS can be increased by adding a local search phase and that $\mathcal{MM}$AS proved to be very valuable in guiding the local search heuristic. In future we want to apply also the more sophisticated Lin-Kernighan heuristic to $\mathcal{MM}$AS to hopefully obtain competitive results with the best algorithms for TSPs. Another issue would be to dynamically adjust β during the run. This seems reasonable as depending on the problem instance, different values for β seem to be best, and higher values for β in the beginning of the algorithm lead to a faster convergence to good tours. To date, $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System seems to be a very promising tool providing an adaptive framework for the solution of combinatorial optimization problems.

Acknowledgments We'd like to thank Marco Dorigo for valuable comments and Christoph Herrmann for careful reading of an earlier draft of this paper.

[7] For random tour construction with the $\mathcal{MM}$AS one simply has to simply set $\alpha = 0$. For β the best value seems to be 5. Note that for $\beta \to \infty$ we obtain the nearest neighbour tour construction heuristic.

References

[1] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed Optimization by Ant Colonies. In *Proceedings of ECAL91 - European Conference on Artificial Life*, pages 134–142. Elsevier Publishing, 1991.

[2] M. Dorigo. *Optimization, Learning, and Natural Algorithms*. PhD thesis, Politecnico di Milano, 1992.

[3] 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*, 26(1):29–41, 1996.

[4] H. B. et.al. Results of the First International Contest on Evolutionary Optimisation. Technical Report TR/IRIDIA/96-18, IRIDIA, Université Libre de Bruxelles, 1996.

[5] M. Fischetti and P. Toth. An Additive Bounding Procedure for the Asymmetric Travelling Salesman Problem. *Mathematical Programming*, 53:173–197, 1992.

[6] L. Gambardella and M. Dorigo. Solving Symmetric and Asymmetric TSPs by Ant Colonies. In *IEEE Conference on Evolutionary Computation (ICEC'96)*. IEEE Press, 1996.

[7] L. M. Gambardella and M. Dorigo. Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. In *Proceedings of the Twelfth Iternational Conference on Machine Learning*, pages 252–260. Morgan Kaufmann, 1995.

[8] G. Reinelt. *The Traveling Salesman: Computational Solutions for TSP Applications*, volume 840 of *LNCS*. Springer Verlag, 1994.

[9] T. Stützle and H. Hoos. Improving the Ant-System: A detailed report on the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. Technical Report AIDA–96–11, FG Intellektik, TH Darmstadt, Aug. 1996.