

Archived in ANU Research repository

<http://www.anu.edu.au/research/access/>

This is the accepted version, published as:

Beer, C, Hendtlass, T. & Montgomery, J. (June 2012)

Improving exploration in Ant Colony Optimisation with antennation.

*Paper presented at the 2012 IEEE Congress on Evolutionary
Computation, Brisbane, Australia, June 10-15, 2012 (pp. 2926-2933).
Piscataway, NJ: IEEE*

© 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Improving Exploration in Ant Colony Optimisation with Antennation

Christopher Beer

SUCCESS¹

Swinburne University of Technology
Melbourne, VIC 3122
Australia

Email: cbeer@swin.edu.au

Tim Hendtlass

SUCCESS¹

Swinburne University of Technology
Melbourne, VIC 3122
Australia

Email: thendtlass@swin.edu.au

James Montgomery

Research School of Computer Science
The Australian National University
Canberra, ACT 0200
Australia

Email: james.montgomery@anu.edu.au

Abstract—Ant Colony Optimisation (ACO) algorithms use two heuristics to solve computational problems: one long-term (pheromone) and the other short-term (local heuristic). This paper details the development of antennation, a mid-term heuristic based on an analogous process in real ants. This is incorporated into ACO for the Travelling Salesman Problem (TSP). Antennation involves sharing information of the previous paths taken by ants, including information gained from previous meetings. Antennation was added to the Ant System (AS), Ant Colony System (ACS) and Ant Multi-Tour System (AMTS) algorithms. Tests were conducted on symmetric TSPs of varying size. Antennation provides an advantage when incorporated into algorithms without an inbuilt exploration mechanism and a disadvantage to those that do. AS and AMTS with antennation have superior performance when compared to their canonical form, with the effect increasing as problem size increases.

Index Terms—Ant Colony Optimization, Travelling Salesman Problem, Mid-Range Heuristic

I. INTRODUCTION

Ant Colony Optimisation (ACO) algorithms, modeled on the behavior of ant colonies, are a class of nature-inspired meta-heuristic search techniques that use a combination of heuristics to influence decisions made by individual agents named ants. The canonical form of ACO uses two heuristics: a long-term heuristic *pheromone* which is analogous to chemical pheromones used by social insects, and a short-term or *local* heuristic.

Antennation is a method of direct interaction observed in social insects where individuals share information directly using their antennae as a communication tool. The *Ecophilla Longinoda* ant species uses antennation, along with other direct interaction techniques, to inform other ants the location of food sources. When antennation is used, other ants are encouraged to follow specific pheromone trails [8].

While typical ACO algorithms use pheromone, representing learning over several generations, this work proposes the use of an analogue of antennation to provide more immediate and direct interactions between artificial ants.

This paper is organised as follows. Section II describes ACO when applied to the TSP. Section III describes three ACO algorithms used in this paper: Ant System (AS), Ant Colony System (ACS) and Ant Multi-Tour System (AMTS). Section IV briefly looks at the balance of exploration and exploitation in ACO and various adaptations used to promote one over the other. Section V introduces the concept of antennation as a direct interaction method and previous adaptations to evolutionary computing in literature. It also details adapting antennation to ACO, specifically the algorithms in Section III, as well as proposing a range of functions used to control the effect of the new adaptations. Section VI provides results on five commonly used symmetric TSP sets: Burma14, Berlin52, Bier127, d198 and pr299. Comparisons are made of the original algorithms to their adaptations as well as between the different algorithms. Section VII discusses the effects of antennation applied to ACO along with possible directions for further research.

II. THE TRAVELLING SALESMAN PROBLEM AND ANT COLONY OPTIMISATION

The Travelling Salesman Problem (TSP) is a non-deterministic polynomial time problem involving finding a minimal cost closed tour of a number of nodes. The TSP comes from the concept of a travelling salesman visiting a given number of cities (nodes) and finding the shortest route connecting all nodes. A TSP has C nodes maximally connected via edges E . The edge E_{ij} connecting nodes i and j has an associated cost η_{ij} and is often the Euclidean distance between nodes.

ACO algorithms are members of the Swarm Intelligence field of metaheuristics and are distinguished by the use of pheromone as a form of indirect communication between software constructs called ants. ACO algorithms were first applied to the TSP due to the similarity between the shortest path to a food source and the shortest complete tour in a TSP, but ACO can and has been applied to a broad range of problems. Solutions to the TSP are developed by depositing attractive pheromone on beneficial edges after a tour of the nodes.

Christopher Beer is a PhD scholar supported by an Australian Postgraduate Award

¹Swinburne University, Centre for Computing and Engineering Software Systems

The amount of pheromone on an edge E_{ij} is termed τ_{ij} and acts as a long-term heuristic by allowing information to be retained between generations. The cost measurement η_{ij} acts as a local short-term heuristic and is unchanged between runs. An important fact to note is that for symmetric TSPs where $\eta_{ij} = \eta_{ji}$, pheromone is unidirectional, i.e., $\tau_{ij} = \tau_{ji}$. Edge selection is influenced by both η and τ . Edge selection is probabilistically determined and provides the stochastic nature of ACO.

The basic process of ACO applied to the TSP is described below:

- 1) Pheromone is initialised on each edge to an initial value τ_0 .
- 2) Ants are placed at their initial (often randomly chosen) nodes.
- 3) Each ant chooses a node to move to with the proviso that it has not already been visited. The node selection choice is non-deterministic and algorithm dependent. This step is repeated until an ant visits all the nodes in the TSP, defined as a tour. The process of all the ants completing a single tour is termed a generation.
- 4) The total distance traveled by each ant is calculated and used to update the current best solution found. The pheromone levels on edges are then updated with the update method also algorithm dependent.
- 5) The algorithm ends if a stopping condition has been reached, otherwise it returns to step 2.

The amount of pheromone deposited on an edge depends on the total distance travelled by the ant after a tour, penalising longer paths and rewarding shorter paths. Mimicking the volatile quality of natural pheromone, τ is also evaporated from all edges after all tours are completed. Over time, commonly travelled paths will be reinforced while less travelled paths will see their pheromone levels eroded. Thus it can be seen that the levels of pheromone on edges will reflect their relative fitness. Eventually the levels of pheromone deposited and evaporated reach equilibrium, the local heuristic is overpowered by the effects of pheromone, ants cease exploring and will converge to travel along the same path.

Two criteria are important when comparing the effectiveness of different ACO algorithms on TSPs: the generations taken for convergence, G , and the length, L , of the corresponding global best solution (S_{gb}). Convergence is deemed to have occurred if there is no change in the best solution after a set number of generations.

III. ACO ALGORITHMS

Edge selection and pheromone update method are the distinguishing aspects of different ACO algorithms.

A. Ant Systems

The original ACO algorithm, known as Ant System [3], was inspired by the foraging behaviour of Argentinean ants. The movement of ants through a TSP is determined by the attractiveness of nodes. At each node, ants construct a 'roulette wheel' of the different probabilities of travelling to each node.

The attractiveness of travelling from node i to node j is given below:

$$A_{ij} = \tau_{ij}^\alpha \cdot \eta_{ij}^\beta \quad (1)$$

τ_{ij} is the level of pheromone on E_{ij} , η_{ij} is the cost of E_{ij} and α and β are user chosen parameters, usually set to 2 and -2 respectively. It is the selection of α and β that determines the balance between exploration and exploitation. For the case that the ant has already visited node j , $P_{ij} = 0$ in all cases.

The actual probability of travelling from node i to node j with C nodes is determined by the *node selection* procedure, as defined below:

$$P_{ij} = \frac{A_{ij}}{\sum_{c=1}^C A_{ij}} \quad (2)$$

Node selection is probabilistically determined and provides the balance between exploration of the search space and exploitation of edges with high τ levels. After all ants have completed a tour and a generation is complete, τ is updated according to the *pheromone update* procedure below:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta_{ij}^k(t) \quad (3)$$

where ρ is the evaporation rate in the range $0 \leq \rho < 1$. Δ_{ij}^k is the amount of pheromone deposited k ants on the edge E_{ij} and is given by Equation 4

$$\Delta^k = \frac{\rho_u}{L^k} \quad (4)$$

where ρ_u , the deposit rate, is a user given value and L^k is the total tour length of ant k .

B. Ant Colony Systems

The Ant Colony System [5] was developed to compensate for problems with the AS algorithm when solving large TSPs. ACS differs from AS in both node selection method and pheromone update procedure. The formula for the attractiveness of an edge is similar to AS with α set to 1. Node selection is now a two-stage process: if a randomly generated number is below a set threshold q , the most attractive node is selected, otherwise the same procedure as in Equation 2 is used. This promotes the exploitation of both inexpensive edges and edges with high levels of pheromone.

The pheromone update procedure is modified in two ways. First, instead of all ants reinforcing their paths, only the global best path experiences pheromone addition, even if no ant has travelled the global best path this tour. Secondly, when an ant traverses a given edge E_{ij} , τ_{ij} is altered according to:

$$\tau_{ij} = (1 - \rho_u) \cdot \tau_{ij} + \rho_u \cdot \tau_0 \quad (5)$$

This *local update* procedure has the effect of lowering pheromone levels on edges traversed by ants within a generation, discouraging other ants from following their exact path. This encourages intra-generational exploration at the expense of the long-term information provided by the pheromone map.

C. Ant Multi-Tour Systems

The previous algorithms share a common feature where ants are replaced after each generation. Ant Multi-Tour System [7] utilises the concept of generational learning with ants internally carrying information from previous tours. In AMTS, ants retain information for a set number of tours mA before being replaced, in particular the number of times each ant has traversed E_{ij} . This information is used to promote exploration with ants becoming increasingly unlikely to follow edges they have previously explored.

AMTS is similar to AS, with the only difference occurring in node selection. Equation 2 is modified slightly to become the following:

$$A_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{F_{ij}} \quad (6)$$

where F_{ij} is a factor derived from $prior_{ij}$, the number of times an ant has traversed E_{ij} . A suitable relationship between F_{ij} and $prior_{ij}$ is given below:

$$F_{ij} = 1 + \sqrt{prior_{ij}} \quad (7)$$

IV. EXPLORATION AND EXPLOITATION

As with any optimisation algorithm, the balance between exploration and exploitation is paramount for ACO algorithms. Exploration refers to how widely an algorithm surveys the search space; in ACO and TSP it is related to the effects of the local heuristic. Exploitation refers to the speed at which the algorithm converges to a local minima and is related to pheromone.

If exploration takes precedence, the algorithm will explore unproductive areas of the search space before reaching a solution; if exploitation is too strong, the algorithm may converge prematurely and produce a poor result [16]. ACO adaptations designed to improve either exploration or exploitation include: resetting pheromone levels [17, 22], introducing multiple colonies [24, 27], using an ‘opposing’ pheromone map [11, 14], hybridising ACO with other metaheuristics [10, 21, 25] and allowing both the best and worst solutions to change pheromone levels during the update process [28].

The majority of alterations to the basic ACO algorithm focus on pheromone intensification of paths and exploitation. A problem with this approach is the loss of diversity and increased chance of premature convergence [6]. ACS encourages exploitation of the best known path while at the same time increasing exploration within a generation by making traversed edges less attractive. AMTS was developed to stimulate exploration by reducing the attractiveness of already traversed edges, increasing exploration at the expense of exploitation.

V. ANTENNATION IN EVOLUTIONARY COMPUTATION

Some species of social insects utilise a method of direct communication, termed ‘antennation’, in which insects rub antennae against each other in rhythmic patterns, sharing various pheromones distinct from those deposited on paths in the environment. This behaviour can communicate a wide range of information: colony hunger levels [1], mating behaviour [19],

nestmate recognition [20] and sexual identification [26]. It has been seen to be able to directly communicate data about the surrounding world between individuals.

Antennation has previously been adapted to swarm intelligence. Trainini, Labella and Dorigo [23] used direct communication between a group of self-assembled robots to explore an area littered with holes. By directly communicating with individuals the whereabouts of nearby holes, the net search effectiveness was increased. Similarly, direct communication was shown to assist multi-robot teams in exploring an obstacle filled arena and increase their search efficiency [2].

A concept similar to antennation is the Meeting ACO algorithm developed by Jun and Wei [9]. Partial solutions of pairs of searching ants are combined midway through a tour, improving exploration while maintaining exploitation of paths. Mavrovouniotis and Yang [12] used direct communication to allow ants to exchange nodes in their tours as long as the result was beneficial. Middendorf, Reischle and Schneck [13] examined sharing good solutions between multiple colonies to improve solution quality.

A. Antennation in ACO

This paper examines the effects of antennation on ACO by implementing a direct communication channel between ants. Antennation in nature is the direct sharing of information between social insects and is distinct from pheromone trails. This paper implements an analogous process where ants meeting at a node share information about the search space. This information is limited in longevity and can be considered a ‘mid-term’ heuristic; a form of intra-generational learning. Information from other ants is gathered by each ant and used to influence decisions within a tour. However, unlike the accumulation of pheromone over generations, this information is erased when a tour is completed and built anew each tour. Also, unlike the short-term heuristic distance, information obtained via antennation is retained throughout a generation and can be shared between ants.

B. Antennation information

There are many options for which information ants can share with one another, from pheromone levels on edges to node attractiveness probabilities. Ants can also share information concerning paths taken and edges traversed. As a significant obstacle to ACO is maintaining diversity at later generations, the concept of antennation was used to develop a mid-range heuristic used as an additional influence towards exploration. Antennation is used as a tool to force diversification of ant paths within a tour, thus providing a mid-term heuristic distinct from pheromone and distance.

In keeping with the design principle of collective intelligence, as well as the interchangeability and simplicity of individuals, each ant can only share its own knowledge about the search space. This form of antennation implemented also stays faithful to the biological analogue of direct communication between insects; only ants located in the same node at the same time share information.

C. Adaptation to ACO

The concept of antenation and direct communication can be implemented in various ways: from ants only sharing their own path, to ensuring shared information is unique by preventing repetitious information being shared. Preliminary research showed that only sharing first-hand information had a negligible effect on either generations til convergence G and global best distance L while ensuring information remained unique proved to be too costly in computational load and violated the premise of anonymity and interchangeability integral to swarm intelligence.

Instead, antenation was adapted to ACO by using a simple strategy involving ants sharing only whether a particular edge had been seen to be travelled, either by the ant itself or any others from which it had received information previously. This form of antenation can be thought of as a coarse memory sharing strategy, instead of sharing the number of ants that have used E_{ij} , an ant will simply increment that edge in its memory ϵ_{ij} . Note that the information is localised to individual ants and is only shared between ants that are on a given node at the same time. This form of antenation can be easily applied to other problems as the memory structure only holds solution components, traversed edges in the case of the TSP.

As an edge can be shared and thus incremented potentially every meeting, e.g., two ants following the same path at the same time, ϵ_{ij} is limited to the total number of ants that have been meet at that point in the tour.

Antenation has been implemented in two ways: ‘bidirectional’ antenation where $\epsilon_{ij} \neq \epsilon_{ji}$ and ‘unidirectional’ antenation where $\epsilon_{ij} = \epsilon_{ji}$. The concept of unidirectional antenation mimics how $\tau_{ij} = \tau_{ji}$ in a symmetric TSP while bidirectional antenation stays closer to the original idea of ants only sharing the paths they have seen taken.

The collected information can be used either for exploration of unseen edges or for exploitation of travelled edges within a generation i.e., it can act as an attractive or repulsive force. As convergence is a significant problem with ACO, antenation was implemented so as to encourage ants to explore edges that other ants have not travelled that generation. Equation 1, which calculates the attractiveness of a given edge and used in the three ACO algorithms, is modified slightly by the inclusion of a multiplicative factor.

$$A_{ij} = \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta} \cdot (1 - f_{type}(A_f, \epsilon_{ij}^k, \gamma^k)) \quad (8)$$

A_f is the Antenation Factor where $0 < A_f \leq 1$, ϵ_{ij}^k is the number of ants that have been seen to traverse E_{ij} by ant k , γ^k is the total number of ants met by the ant k . f_{type} is the antenation control function and determines the relative influence of ϵ_{ij} and N .

D. Control functions

Three different functions were used to examine how scaling the multiplicative factor in Equation 8 affects average distance and generations until convergence. Since antenation is a simple multiplicative factor, the extent to which it affects the attractiveness of a given edge is important. Three different

TABLE I
PARAMETER SETTINGS

	AS	AMTS	ACS
α	2	2	1
β	2	2	2
τ_0	0.1	0.1	$2.2e^{-5}$
ρ	0.1	0.1	0.1
ρ_u	0.5	0.5	0.1
q	n.a	n.a	0.9
mA	n.a	6	n.a

scaling factors were implemented to examine A_f and its relationship with ϵ .

$$\begin{aligned} \text{Boolean } f_{type}(\epsilon_{ij}, \gamma) &= 0 \text{ if } \epsilon_{ij} = 0 \text{ else } AntF_{ij} = A_f \\ \text{Linear } f_{type}(\epsilon_{ij}, \gamma) &= A_f \cdot \frac{\epsilon_{ij}}{\gamma} \\ 1/\epsilon f_{type}(\epsilon_{ij}, \gamma) &= \frac{A_f \cdot \epsilon_{ij}}{1 + A_f \times \epsilon_{ij}} \end{aligned}$$

Boolean is the simplest control function implementation: if an ant has information that any ants have traversed edge E_{ij} , the attractiveness of that edge is reduced immediately. Boolean has the advantage of quickly altering the attractiveness of edges but does not increase when the number of ants that have actually traversed the edge increase.

Linear uses the ratio of γ and ϵ to control the effect of antenation i.e., as γ increases during a tour, f_{type} will decrease unless ϵ_{ij} increases in proportion. Linear seeks to provide a level of fine control in comparison to Boolean.

Unlike the previous two control functions which are bounded by A_f , $1/\epsilon$ is bounded by 1. It has a faster response to change in comparison to Linear while being more sensitive than Boolean.

VI. COMPUTATIONAL RESULTS

To assess the performance of antenation when applied to the three ACO algorithms, experiments were run using five TSP data sets taken from the TSP-LIB [18]: Burma14, Berlin52, Bier127, d198 and pra299. Based on Montgomery, Randall and Hendtlass [15], the parameters in Table I were used. The number of ants N was set to 10 for Burma14, 25 for Berlin52 and 40 for Bier127, d198 and pra299.

The 2-opt algorithm was used as a local search technique at the end of every tour. The canonical form of 2-opt in which two nodes are selected at random and swapped in order, keeping the better solution [4], was slightly modified for AACO. An exponential distribution was used to select nodes, biasing towards selecting nodes closer in a given path rather than further away. This has the effect of preferring small changes rather than large changes in solution, a key factor in local search.

A run was considered to have converged if there had been 100 generations without improvement. Thus G was calculated by the number generations at halt minus 100. A_f was varied from 0 to 0.95 in increments of 0.05. Performance was analysed using two criteria: generations until convergence (G) and best solution length (L). 200 runs were completed at each setting and the results averaged.

TABLE II
 A_f VERSUS G FOR BURMA14

A_f	0.2	0.4	0.6	0.8	0.95
AS	53.5	53.5	53.5	53.5	53.5
Boolean	1.1/-0.1	1.2/2.6	1.7/5.1	3.2/9.8	9.8/25.4
Linear	1.1/0.2	0.3/0.1	0.2/1.7	1.2/2.3	1/4.7
$1/\epsilon$	-0.2/1.7	0.6/3.6	1.4/4.1	2.4/5.5	2.8/4.6
ACS	17.2	17.2	17.2	17.2	17.2
Boolean	-0.4/-0.7	-0.3/-0.3	-0.7/-0.3	0.2/3.4	9.2/18.9
Linear	-0.7/-0.4	0/-0.5	-0.4/0	-0.4/0.7	0.6/5.8
$1/\epsilon$	-0.1/-0.5	0.8/1.6	12/-0.4	0.1/0.1	0.5/-0.1
AMTS	59.4	59.4	59.4	59.4	59.4
Boolean	0.5/1.3	0.3/2.1	2.2/4.4	4.8/12.3	11.6/27.2
Linear	-0.3/0.4	0.3/0.3	1/0.8	0.5/2.7	0.9/5.4
$1/\epsilon$	0.4/1.5	1.7/3.6	2.2/3.6	2.2/4.6	3.3/5.7

TABLE III
 A_f VERSUS L FOR BURMA14

A_f	0.2	0.4	0.6	0.8	0.95
AS	3360	3360	3360	3360	3360
Boolean	-0.8/1.9	-0.5/1.5	1.5/1.9	3.7/4.3	6.3/7.7
Linear	-1.9/1.1	1/1.3	0.3/0.6	1.6/3	1.1/1
$1/\epsilon$	2/1	1.2/1.2	-0.8/2.6	1.7/2.3	2.5/4.8
ACS	3361.3	3361.3	3361.3	3361.3	3361.3
Boolean	-3.7/-4.9	-1.2/-11.5	-4.7/-14.9	-10.9/-16.9	-17.1/-23.5
Linear	1.6/-4.6	2/-4.7	-0.47/-9.7	-6.2/-13.4	-11.9/-24.4
$1/\epsilon$	-2.4/-10	-1.5/-11.9	-4.4/-13.1	-5.1/-17	-5.7/-17.1
AMTS	3363	3364	3365	3366	3367
Boolean	-0.5/0.4	1/0	0.8/3.4	2.7/1.2	7.6/5.5
Linear	1.8/0.1	0.7/1.5	0.5/0.5	0.8/2.6	1/1.3
$1/\epsilon$	1.1/-0.2	0.1/0.7	0/1.8	2.3/1.9	1.1/3.4

TABLE IV
 A_f VERSUS G FOR BERLIN52

A_f	0.2	0.4	0.6	0.8	0.95
AS	76.3	76.3	76.3	76.3	76.3
Boolean	3.2/2.4	6/7.7	12.5/12.7	31.3/28.5	75.1/91.1
Linear	1.9/2.8	3/3.8	4.2/5.6	6.6/9.6	10.9/17
$1/\epsilon$	12.2/12	19.6/20.6	25.5/27.7	30.3/32.7	35.3/37
ACS	55.7	55.7	55.7	55.7	55.7
Boolean	2.7/2.7	12.4/6.8	15.2/18.4	2.4/11.2	6.9/12.1
Linear	1.6/2.4	2.3/4.7	9.4/10	11.4/12.5	10.5/8.4
$1/\epsilon$	12.2/12.5	9.5/13.1	6.7/13.3	8.5/16	6.4/13.1
AMTS	90.9	90.9	90.9	90.9	90.9
Boolean	3.9/3	8.7/8.2	18.8/17.2	38/35.3	47.4/73
Linear	0.8/2.2	3/4.4	6.2/8.7	9.4/14.6	15.9/22.1
$1/\epsilon$	16/15.2	25.6/27	30.1/32	33.2/40.7	38.6/44.7

TABLE V
 A_f VERSUS L FOR BERLIN52

A_f	0.2	0.4	0.6	0.8	0.95
AS	7757.9	7757.9	7757.9	7757.9	7757.9
Boolean	-13.9/13.7	-12.6/-25	-21.1/-27.6	-39/-45.9	169/-15.2
Linear	-2.8/-10.7	-9.9/-11	-18.8/-18.2	-20.3/-29.4	-21.6/-40.4
$1/\epsilon$	-15.1/-13.8	-3.4-17.5	-2.2/-24.5	3.7/-16.6	13.2/-10.5
ACS	7759.2	7759.2	7759.2	7759.2	7759.2
Boolean	-4.7/0.8	-12.3/-12.3	42.3/63.8	604/555	1397/1820
Linear	-8.8/1.6	-13.3/-10.5	-17.1/-12.3	-11.2/1.9	37.1/72.2
$1/\epsilon$	66.9/99	255/233	408/367	522/491	610/600
AMTS	7737.2	7737.2	7737.2	7737.2	7737.2
Boolean	-5.5/-10.5	-7.7/-19.3	-24.8/-28	-4.2/-29.5	532/232
Linear	-1.4/-3.7	-1.7/-7.9	-9.7/-14	-9.4/-21.5	-16.2/-28
$1/\epsilon$	-7.6/-10.2	1.1/-16.6	20.7/-5.2	44.6/-1.5	63.2/9.2

TABLE VI
 A_f VERSUS G FOR BIER127

A_f	0.2	0.4	0.6	0.8	0.95
AS	85.6	85.6	85.6	85.6	85.6
Boolean	6.1/7.4	12.2/18.9	15.5/25.1	22.7/34.4	48.3/53.3
Linear	1.5/3.4	-2.5/8	3.4/13.4	4.0/19.6	14.5/15.8
$1/\epsilon$	3.4/4.5	7.5/6.2	9.4/10.4	12.3/14.8	34.8/41.4
ACS	152.7	152.7	152.7	152.7	152.7
Boolean	-9.4/1.4	3.5/8.4	6.9/10.6	29.2/22.4	42.1/48.4
Linear	2.2/-4.4	-2.4/5.3	5.4/11.6	7.5/15.4	19/12.5
$1/\epsilon$	3.4/5.3	4/11.4	11.4/17.6	18.5/22.4	36.3/45.4
AMTS	91.7	91.7	91.7	91.7	91.7
Boolean	11/3.5	8.4/13.8	17.8/12.6	25.2/30.2	71/92.9
Linear	-2.6/-0.7	9.1/1.3	11.9/-0.3	17.3/25.2	30.2/28.7
$1/\epsilon$	1.3/4.8	60.3/24.8	25.1/39	30.2/30.9	48.6/71.2

Results are given in 10 tables: Tables II,IV,VI,VIII,X show the net change in G in respect to A_f for the five problems, Tables III,V,VII,IX,XI show the net change in L in respect to A_f for the five problems.

Results from bidirectional and unidirectional antenation is given by the left hand and right-hand number respectively and is calculated by subtracting the results of the canonical form. The results of the canonical algorithm forms are given at the top of each sub-table. For brevity, only results for $A_f = [0.2, 0.4, 0.6, 0.8, 0.95]$ are displayed.

A. Discussion

AACO has different effects depending on the size of the problem and the ACO algorithm type. A common result in all problems is an increase in G as A_f increases, This is likely due to antenation acting to delay convergence and maintaining diversity at higher values of A_f . Instead of converging, antenation encourages ants to explore variants on commonly traversed arcs and thus increases the number of generations til convergence.

It can be seen from Table II and III that altering A_f and the

TABLE VII
 A_f VERSUS L FOR BIER127

A_f	0.2	0.4	0.6	0.8	0.95
AS	126232	126232	126232	126232	126232
Boolean	-11/-78	-22/-78	-54/-86	-108/-141	1661/1766
Linear	-26/-13	-10/-44	-29/-37	-54/-68	27/-24
$1/\epsilon$	-21/-51	-54/-68	-59/-79	-93/-81	540/688
ACS	122898	122898	122898	122898	122898
Boolean	395/744	603/519	803/819	1036/1100	1682/1523
Linear	107/12	205/86	148/45	199/119	532/583
$1/\epsilon$	533/776	416/713	585/892	681/993	1106/1119
AMTS	123522	123522	123522	123522	123522
Boolean	-38/-98	-89/140	-132/-210	-154/-223	1181/1750
Linear	-1/21	-21/-2	-36/-12	-21/-23	55/22
$1/\epsilon$	-12/-16	-15/-28	-19/-33	-31/-67	146/500

TABLE VIII
 A_f VERSUS G FOR D198

A_f	0.2	0.4	0.6	0.8	0.95
AS	118.7	118.7	118.7	118.7	118.7
Boolean	1.5/4.6	4.3/2.3	10.3/19.1	19.8/26.3	75.4/69.8
Linear	-6.4/5.7	0.4/7.3	1.9/12.2	10.9/10.1	24.3/33.6
$1/\epsilon$	6.3/5.3	9.4/7.4	7.5/9.6	16.3/21.5	41.2/32.3
ACS	83.1	83.1	83.1	83.1	83.1
Boolean	7.8/5.2	13.1/12.7	23.3/18.0	32.5/30.1	45.2/39.0
Linear	1.1/3.2	3/4.8	7.3/3.3	17.4/8.2	19.9/15.6
$1/\epsilon$	8.1/4.2	10/7.5	21.2/27.4	24.2/32.1	30.6/52.7
AMTS	134.6	134.6	134.6	134.6	134.6
Boolean	7.1/6.3	10.8/5.1	14.1/7.6	15.4/21.4	24.6/27
Linear	4.2/0.8	1.5/1.2	0.3/8.4	3.8/14.9	6.9/20.3
$1/\epsilon$	3.3/4.8	-0.9/16.8	11.7/14.3	3.1/22.8	12/29.4

TABLE IX
 A_f VERSUS L FOR D198

A_f	0.2	0.4	0.6	0.8	0.95
AS	17934	17934	17934	17934	17934
Boolean	-32/-42	-56/-46	-70/-102	-114/-133	1009/1709
Linear	68/-13	-31/-33	-10/-29	-46/-48	22/-61
$1/\epsilon$	-43/-16	-40/-35	-57/-42	-52/-63	608/269
ACS	17621	17621	17621	17621	17621
Boolean	60/1	122/180	250/281	332/332	910/590
Linear	-14/-6	47/96	143/209	166/399	263/602
$1/\epsilon$	22/34	90/403	213/534	311/421	711/1339
AMTS	17610	17610	17610	17610	17610
Boolean	-7/-25	-70/-71	-46/-115	-257/-84	1538/947
Linear	31/31	-7/-18	34/2	132/78	201/170
$1/\epsilon$	-24/-23	-57/-18	-63/-23	-168/-60	106/94

control function has a relatively weak effect on performance on the smaller problem Burma14. AS, AMTS are little affected by antennation, both unidirectional and bidirectional, with little variation in L regardless of type of control function or value of A_f . ACS demonstrates a net decrease in L as A_f increases, regardless of what control function is being used. This could be due to antennation providing a beneficial shift towards exploration or simply a sampling artefact as the difference in L is minimal. The Boolean control function demonstrates the greatest improvement, followed closely by Linear and $1/\epsilon$.

TABLE X
 A_f VERSUS G FOR PR299

A_f	0.2	0.4	0.6	0.8	0.95
AS	112.6	112.6	112.6	112.6	112.6
Boolean	3.6/5.5	4.7/3.8	11/8.3	18.3/14.7	51.7/61.4
Linear	-2.4/-0.4	-5.8/1.9	3.5/6.8	6.8/8.5	11.9/10.7
$1/\epsilon$	2.7/3.6	1.6/2.8	4.5/0.5	11.5/8.7	17.6/15.3
ACS	89.4	89.4	89.4	89.4	89.4
Boolean	2.7/2.7	12.4/6.8	15.2/18.4	2.4/11.2	6.9/12.1
Linear	1.6/2.4	2.3/4.7	9.4/10	11.4/12.5	10.5/8.4
$1/\epsilon$	12.2/12.5	9.5/13.1	6.7/13.3	8.5/16	6.4/13.1
AMTS	131.7	131.7	131.7	131.7	131.7
Boolean	2.1/1.4	12.1/14.4	/17.2	21.3/24.9	31.1/56
Linear	-3/5.5	4.5/4	10.2/10.8	12.6/24.1	14.3/23.5
$1/\epsilon$	10.2/24.1	18.8/30	27.4/30.2	40.2/54.4	45.5/60.4

TABLE XI
 A_f VERSUS L FOR PR299

A_f	0.2	0.4	0.6	0.8	0.95
AS	57775	57775	57775	57775	57775
Boolean	-74/-345	-544/-764	-926/-1140	-1972/-2134	5267/54
Linear	-20/-57	-110/-147	-246/-343	-443/-743	-554/-435
$1/\epsilon$	-66/-246	-/-770	-220/-679	-789/-1024	1103/-54
ACS	55443	55443	55443	55443	55443
Boolean	-4.7/0.8	-12.3/-12.3	42.3/63.8	604/555	1397/1820
Linear	-8.8/1.6	-13.3/-10.5	-17.1/-12.3	-11.2/1.9	37.1/72.2
$1/\epsilon$	66.9/99	255/233	408/367	522/491	610/600
AMTS	56320	56320	56320	56320	56320
Boolean	145/-499	-359/-379	-695/-1110	-921/-1219	3172/3701
Linear	20/3	-10/-43	-58/-88	-135/-244	-213/-344
$1/\epsilon$	-103/-54	-53/-155	-78/-88	-54/43	1034/431

When antennation is used on the more complex problems Berlin52, Bier127, d198 and pr299, AS and AMTS show a strong correlation between A_f and L . This is readily apparent for the more sensitive control function Boolean and $1/\epsilon$, less so for Linear. This correlation is shown clearly in the range $0 < A_f \leq 0.8$ for Berlin52 and pr299 with unidirectional antennation having a greater effect on L than bidirectional antennation. This can be attributed to the increased amount of information being shared throughout a tour preventing premature convergence. Figures 1 and 2 show the correlation between A_f and L for AMTS and AS is $> 0.95\%$ for these problem instances.

As A_f approaches 1, the incentive for exploration becomes more of a hindrance than a benefit. Instead of consolidating heavily travelled edges (inferred to be efficient), antennation appears to force ants to explore costly arcs and finally forces the algorithm to converge to a poor solution. What must be noted is that the ants are still exploring the problem space when A_f approaches 1 and the convergence threshold is triggered. However, pheromone, the long-term heuristic that encourages exploitation, is now overwhelmed by antennation, the mid-term heuristic.

ACS is poorly affected by antennation when applied to the larger problems. Unlike in Burma14, antennation increases both L and G , especially for $1/\epsilon$ and Boolean. This is most likely due to ACS having an exploration mechanism where ants update pheromone levels within a generation. By deterring ants from following edges that have been traversed within a generation, the exploitation aspect of ACS is weakened, making good solutions much harder to achieve. Antennation interferes with the balance between exploration and exploitation in ACS in an unfavourable way.

VII. CONCLUSION

The implementation of antennation proposed in this paper has been shown to have positive effects on the problems analysed. It appears that Burma14 is too small for an effect on exploration to occur for AS and AMTS whereas for the larger problems, AACO increases the performance of AS and AMTS, well past the performance of the original implementations. AS comes very close to outperforming AMTS as made clear

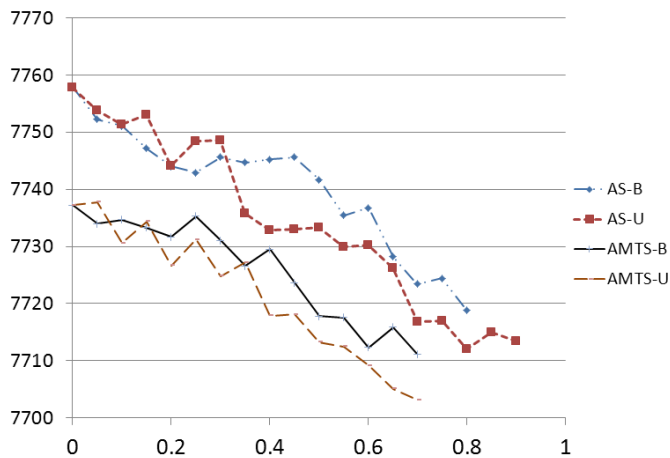


Fig. 1. $L(\text{vertical})$ vs A_f with AMTS and AS for Berlin52

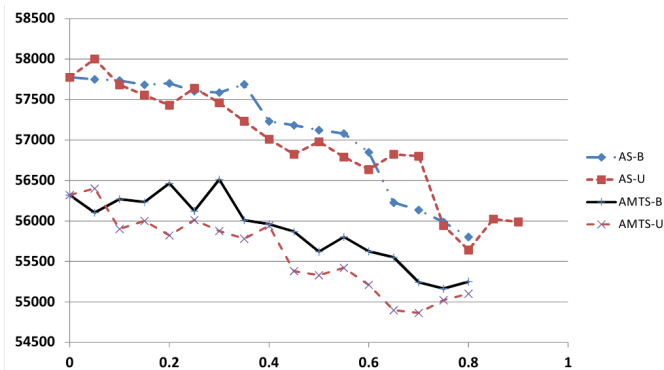


Fig. 2. $L(\text{vertical})$ vs A_f with AMTS and AS for pr299

in Figure 1 and Figure 2; AS shows greater improvement in performance for larger values of A_f when compared to AMTS.

The opposite is true when adapting antenation to ACS. ACS shows enhanced exploration for Burma14 with a strong correlation with A_f and L . This is likely due to the trivial nature of Burma14; ACS benefits from the increased exploration but its exploitation mechanisms have a greater effect and prevent any loss of pheromone intensification. When the size of the problem space is increased as for the larger problems, antenation and ACS do not synergize well and the net effect is a strong negative impact on the balance between exploration and exploitation.

By using control mechanisms that are more sensitive to the numbers of ants seen to traverse an arc have a greater effect especially when using control functions such as Boolean. The simplest implementation of antenation, using a switch statement, appears to be most effective out of the proposed control functions. By making antenation unidirectional, the effect on both generations until convergence, and length of best solution is increased again. This is due to the increased probability that f_{type} will be a non-zero value, influencing the node selection process.

Antenation can provide a significant push towards exploration of the search space when applied to ACO algorithms that

have issues with convergence. This results in better solutions in terms of average distance but worse performance in terms of generations required due to the increased exploration. This is readily apparent in the larger TSPs, as the antenation mechanism requires multiple meetings to build up a memory of traversed edges. These results indicate that as the number of nodes increases, the effect of antenation becomes stronger. Intuitively, if the total number of nodes is increased and/or more ants are introduced, the probability that, during a tour, ants will meet other ants with beneficial knowledge of traversed possible arcs increases, maintaining diversity in node selection.

It is clear that by encouraging ants not to use the partial paths of others, there is a beneficial effect on exploration and this effect can be controlled by changing A_f .

VIII. FURTHER WORK

As ACO algorithms are weighted towards exploration in the initial stages and exploitation in the end stages, controlling when antenation is active could be of most use in the mid-stage to avoid premature convergence.

As antenation is an intra-generational learning technique, applying antenation to larger TSPs could allow a greater level of information to be built up over time, increasing its effectiveness. Other ways to increase the effectiveness of antenation could include increasing the number of ants in a simulation or retaining information between tours, similar to AMTS.

Investigating the mechanics of antenation itself could prove useful as this paper has only examined a basic method of intra-generational communication. Other information that ants could gather and share include: pheromone levels on arcs, partial path quality, distance over remaining nodes etc. Also, exploitation could be promoted over exploration. A switch between exploration and exploitation could be made when a threshold is passed.

This paper has proposed a form of antenation and applied it successfully to the TSP but the underlying mechanism of sharing information concerning partial solutions can be applied to any problem. Investigating the effects of antenation on other problem spaces should prove fruitful.

REFERENCES

- [1] S. Cabe, W. Farina, and R. Josens. Antenation of nectar-receivers encodes colony needs and food-source profitability in the ant *Camponotus mus*. *Insectes Sociaux*, 53:356–361, 2006.
- [2] O. Çayırpunar, V. Gazi, B. Tavli, E. Cervera, U. Witkowski, and J. Penders. Experimental study on the effects of communication range on cooperative robotic search in complex environments. In *Second International Conference on Robot Communication and Coordination*, pages 1–6, 2009.
- [3] A. Coloni, M. Dorigo, and V. Maniezzo. Distributed optimization by ant colonies. In *European Conference on Artificial Life*, pages 134–142, 1991.

- [4] G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6:791–812, 1958.
- [5] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–66, 1997.
- [6] M. Dorigo and T. Stitzle. *Ant Colony Optimization*. MIT Press, Cambridge, 2004.
- [7] T. Hendtlass. TSP optimisation using multi tour ants. In *Innovations in Applied Artificial Intelligence*, volume 3029 of *Lecture Notes in Computer Science*, pages 523–532. Springer Berlin / Heidelberg, 2004.
- [8] B. Hlldobler and E. Wilson. *Journey to the Ants: A Story of Scientific Exploration*. Belknap Press, July 1998.
- [9] Z. F. Jun and G. Wei. Meeting ant colony optimization. In *IEEE International Symposium on Knowledge Acquisition and Modeling Workshop*, pages 972–975, December 2008.
- [10] Q. Ling, C. Yixin, L. Jianli, C. Ling, and G. Jing. A diversity guaranteed ant colony algorithm based on immune strategy. volume 2, pages 217–223, 2006.
- [11] A. Malisia. Improving the exploration ability of ant-based algorithms. *Studies in Computational Intelligence*, 155:121–142, 2008.
- [12] M. Mavrovouniotis and S. Yang. Ant colony optimization with direct communication for the traveling salesman problem. In *2010 UK Workshop on Computational Intelligence (UKCI)*, pages 1–6, September 2010.
- [13] M. Middendorf, F. Reischle, and H. Schmeck. Multi colony ant algorithms. *Journal of Heuristics*, 8:305–320, 2002. 10.1023/A:1015057701750.
- [14] J. Montgomery and M. Randall. Anti-pheromone as a tool for better exploration of search space. In *Proceedings of the Third International Workshop on Ant Algorithms*, ANTS '02, pages 100–110, London, UK, 2002. Springer-Verlag.
- [15] J. Montgomery, M. Randall, and T. Hendtlass. Automated selection of appropriate pheromone representations in ant colony optimisation. *Artificial Life*, 11:269–291, 2005.
- [16] Y. Nakamichi and T. Arita. Diversity control in ant colony optimization. *Artificial Life Robot*, 7:198–204, 2004.
- [17] M. Randall and E. Tonkes. Intensification and diversification strategies in ant colony system. *Complexity International*, 9, 2002.
- [18] G. Reinelt. *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer-Verlag, Berlin, Heidelberg, 1994.
- [19] R. Romani, N. Isidoro, P. Riolo, F. Bin, A. Fortunato, S. Turillazzi, and L. Beani. A new role for antennation in paper wasps (hymenoptera, vespidae): antennal courtship and sex dimorphic glands in antennomeres. *Insectes Sociaux*, 52:96–102, 2005.
- [20] T. Seeley. *The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies*. Belknap Pr, 1995.
- [21] B. Shuang, J. Chen, and Z. Li. Study on hybrid ps-aco algorithm. *Applied Intelligence*, pages 1–10, 2009.
- [22] E. G. Talbi, O. Roux, C. Fonlupt, and D. Robillard. Parallel ant colonies for the quadratic assignment problem. *Future Generation Computer Systems*, 17:441–449, 2001.
- [23] V. Trianni, T. H. Labelle, and M. Dorigo. Evolution of direct communication for a swarm-bot performing hole avoidance. In *Ant Colony, Optimization and Swarm Intelligence*, volume 3172, pages 174–197. Springer Berlin / Heidelberg, 2004.
- [24] H. Xue, P. Zhang, and L. Yang. A multiple ant colonies optimization algorithm based on immunity for solving tsp. pages 289–293, 2010.
- [25] X. You, S. Liu, and Y. Wang. Quantum dynamic mechanism-based parallel ant colony optimization algorithm. *International Journal of Computational Intelligence Systems*, 3:101–113, 2010.
- [26] E. Zefa, L. d. P. Martins, and N. Szinwelski. Complex mating behavior in *Adelosgryllus rubricephalus* (orthoptera, phalangopsidae, grylloidea). *Iheringia. Série Zoologia*, 98:325–328, September 2008.
- [27] P. Zhang and J. Lin. An adaptive heterogeneous multiple ant colonies system. volume 1, pages 193–196, 2010.
- [28] Y. Zhang, H. Wang, Y. Zhang, and Y. Chen. Best-worst ant system. In *2011 3rd International Conference on Advanced Computer Control*, pages 392–395, 2011.