ANT COLONY ALGORITHM (ACA)

Prof. G. Panda, FNAE, FNASc.
Indian Institute of Technology, Bhubaneswar
Email : ganapati.panda@gmail.com / gpanda@iitbbs.ac.in

Presentation Flow

- Introduction
- The Traveling Salesman Problem
- Ant Colony Algorithm (ACA)
- Genetic Algorithm (GA)
- Ants Net Algorithm
- Numerical Examples
- Conclusions
- References

Introduction

Soft computing

❑ Inspired by the biological computational methods and nature's problem solving strategies.

❑ **Includes**

- **Artificial Neural Networks (ANN) – Model of Neurons of the brain**,
- **Genetic Algorithm** - Evolutionary Computational model,
- **Fuzzy logic** - Linguistic method.

❑ ***Soft Computing = Computational Intelligence.***

Introduction Contd…

Ant Colony Algorithm (ACA)

- **A component of Swarm Intelligence.**
- **Used for Optimization.**

Application Areas

- ✓ Vehicle Routing,
- ✓ Sequential Ordering,
- ✓ Graph Coloring,
- ✓ Routing Communication Networks etc.

Introduction Contd…

What do Foraging ants and long distance truck drivers have in common??

Both want to deliver the loads as quickly as possible.

But while ants invariably take the shortest path to their destinations, human counterparts often choose a more circuitous route.

So, thanks to the new algorithm ACA (Ant Colony Algorithm) that imitates insect behavior, trucks can be as efficient as ants.

ACA **-** proposed by **Dorigo and his colleagues** as a multi-agent approach to solve difficult combinatorial optimization problem such as **Traveling Salesman Problem (TSP), Quadratic Assignment Problem (QAP).**

Introduction Contd…

Insects that live in colonies - ants, bees, wasps, termites have long fascinated everyone from naturalists to artists.

Each insect in a colony seems to have its own agenda and yet the group as a whole is highly organized.

Ants' behavior is unsophisticated and they collectively perform complex tasks. Ants have highly developed sophisticated sign based stigmergy.

Ants deposit a trail of *pheromones* - ***chemicals whose smell can inform or influence the behavior of other ants*** - along the route they travel in search of food.

Introduction Contd…

When a food source is found, the ant that discovered it communicate this information to its peers, who thus follow that insect's ***pheromone trail***.

As more and more ants travel to the food source the pheromone track becomes thicker and thicker attracting more and more ants who in turn deposit their own pheromone and so on.

When *confronted to the obstacle* (**Fig.1)** on the preferred path, the ants ***quickly switch to the next most efficient line to the food***.

A high level of pheromone on the **right path** gives the ant stronger stimulus and has higher probability to turn right.

Introduction Contd…

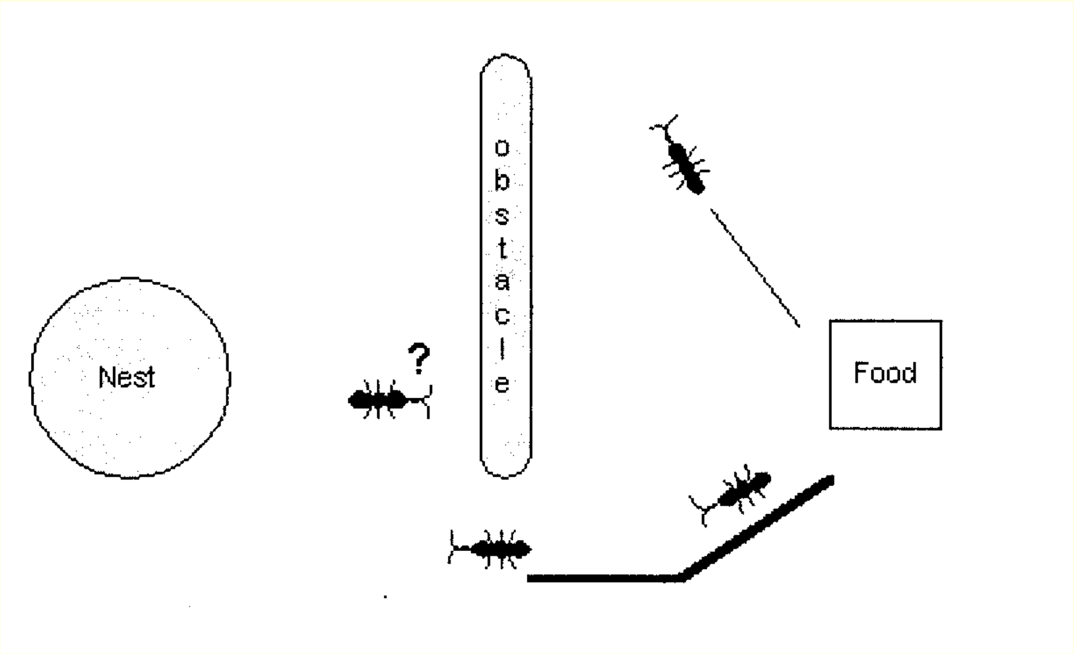

Fig. 1: Path of Ants when there is obstruction

Introduction Contd…

*Consider **A** is the <u>food source</u>, **E** is the <u>nest</u> and an <u>obstruction</u> **HBCD**. The distances of various points are shown in **Fig.2.***

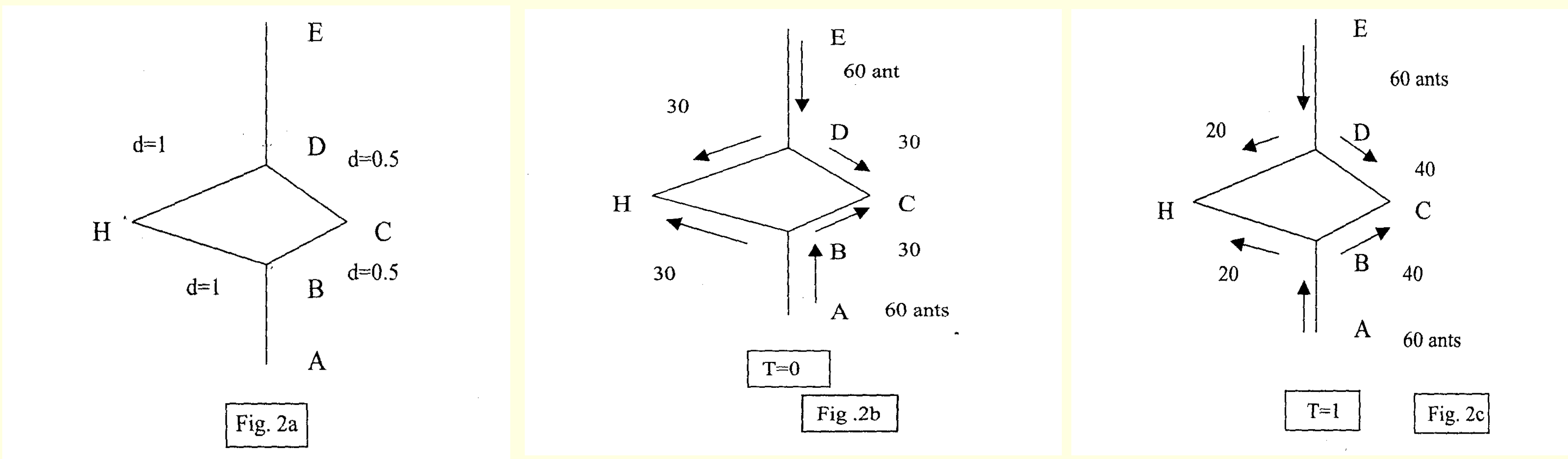

Fig. 2: Pheromone Trail of Ants

Introduction Contd…

At **T = 0**,

60 ants at **B** and 60 at **D** on the average 30 ants from each node will go towards **H** and 30 towards **C**.

At T=1,

The new 60 ants that come to **B** from **A** find a trail of intensity 30 on the path that leaves to **H** laid by 30 ants that went the way from **B** and trail of intensity 30 on path to **C** obtained as sum of trail laid by 30 ants that went all way from **B** and 30 ants coming from **D** via **C**.

*The probability of choosing the path is based so that expected number of ants going towards **C** is double towards it.*

The process continues like all the ants will eventually choose the shortest path.

The Traveling Salesman Problem (TSP)

The ***Traveling Salesman Problem*** *(TSP) is a well known in optimization.*

TSP is the problem of a salesman who wants to find, starting from his hometown a shortest possible trip through a given set of customer cities and to return to his hometown.

The weighted graph,

$$\boldsymbol{G = (N, A)}$$

N = set of nodes representing the cities.

A = set of arcs fully connecting the nodes N.

Each are assigned a value d_{ij}, which is the length of the arc $(i, j) \in A$ in the distance between cities i, j with $(i, j) \in N$.

The TSP is the problem of finding a minimal length Hamiltonian circuit of the graph, where an Hamiltonian circuit is a closed tour visiting exactly once each of the $n = |N|$ nodes of G.

TSP Contd…

For **symmetric TSPs**, the distances between the cities are independent of the direction of traversing the arcs, that is $d_{ij}=d_{ji}$ for every pair of nodes.

And in **asymmetric TSP (ATSP)** at least for one pair of node i, j,

we have, $d_{ij} \neq d_{ji}$.

The TSP in a Nutshell

For $\boldsymbol{n}$ **cities to visit**, let X_{ij} has the variable that has value 1 if the salesman goes from city i to city j and the value 0 if the salesman does not go from city i to city j. Let d_{ij} is the distance from city i to city j.

TSP Contd…

Maximize the linear objective function

$$\sum^{n} \sum_{j=1}^{n} X_{ij} d_{ij} \qquad (1)$$

Subject to

$$\sum_{i=1, i \neq j}^{n} X_{ij} = 1$$ for each $j=1, 2, \dots n$ (linear constraint) (2)

$$\sum_{i=1, i \neq j}^{n} X_{ij} = 1$$ for each $i=1, 2, \dots n$ (linear constraint (3)

$$\sum X_{ij} = 0$$ for 1 for all i and j (integer constraint) (4)

Ant Colony Algorithm (ACA)

Procedure:

In **ACA**, the solution is built by the ants and the algorithm executes t_{max} iterations and during each iteration 'm' ants build a tour executing 'n' steps in which probability decision rule is applied. In general when the ant in node 'i' chooses to node 'j' and the arc $(i\text{-}j)$ is added to the tour under construction. This is repeated till the ant completes its tour.

In any 't^{th}' iteration, the amount of pheromone $\tau_{ij}(t)$ associated to arc $(i\text{-}j)$ is intended to represent the learned desirability of choosing city 'j' when in city 'i'.

ACA Contd…

If the ant K has visited some cities already, this memory is called "*tabu list*". The ant decision table $[a_{ij}(t)]/\chi_i$ of node 'i' is given by

$$a_{ij}(t) = \frac{(\tau_{ij}(t))^{\alpha}(\eta_{ij}(t))^{\beta}}{\sum_{i \in \chi_i} (\tau_{ij}(t))^{\alpha}(\eta_{ij})^{\beta}} \qquad (5)$$

$\tau_{ij}(t)$ = amount of pheromone trail on the arc (i, j) at time 't'.

$\eta_{ij} = 1/d_{ij}$, d_{ij} = distance between nodes i to j.

$d_{ij} = d_{ji}$ (symmetric) and $dij \neq dji$ (un-symmetric).

χ_i = number of neighbors of node 'i'.

α, β = control parameters.

ACA Contd…

The probability with which ant 'K' chooses to go from city 'i' to city j while building its tour at the t^{th} iteration is

$$p_{ij}^{k}(t) = \frac{a_{ij}(t)}{\sum_{l \in \chi_i^k} a_{il}(t)} \qquad (6)$$

where

$\chi_i^k \subseteq \chi_i$ = set of node in the neighbor of i that 'K' has not visited 'yet'.

ACA Contd…

After ants have completed the tour, pheromone evaporation of arcs is triggered and then each ant 'K' deposits a quantity of pheromone $\Delta\tau_{ij}^{k}(t)$ on each arc it has used as

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} 1/L^{k}(t) & \text{if } i, j \in T^{k}(t) \\ 0 & \text{if } i, j \notin T^{k}(t) \end{cases} \qquad (7)$$

$T^{k}(t)$ = tour done by ant 'K' at iteration 't'.

$L^{k}(t)$ = its length.

It is clear from (3) that the shorter the tour the greater the pheromone deposited. Pheromone evaporation is incorporated as

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \qquad (8)$$

ACA Contd…

Where

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \qquad (9)$$

ρ = pheromone decay coefficient.

'm' = total number of ants.

'n' = total number of cities.

$\alpha = 1$ and $\beta = 2$ to 5.

Genetic Algorithm (GA)

- **Genetic Algorithm (GA)** is a computational model based on natural evolution originally developed by **Holland**.
- GA are attractive classes of computational models that mimic natural evolution to solve the problems in wide variety of domains.
- Pioneering work of Holland, Goldberg, Dejong Grefenstette, Davis and Muhlenbein and others were fueling the spectacular growth of GA.
- It is used as a ***powerful optimization tool***.
- A system to be optimized is *represented by a binary string*, *which encodes the parameters of the system.*
- A population of strings with initial random parameters is used.
- Genetic Algorithm combines the **Darwinian theory of survival of the fittest procedure.**

GA Contd…

- A number of generations are simulated with operators representing the major elements of evolution such as competition, fitness based selection, recombination such as cross over and mutation.
- Inversion, duplication, segregation, dominance, intra chromosomal duplication, translocation, speciation, migration, sharing are also used.

In the TSP instead of binary coding the real coding (Fig.3).

Once the shortest path is obtained from **ACA** the nodes of all the cities are written in order and two cross-sites are selected.

For the example, two cross sites are 2 and 5.

GA Contd…

If the m^{th} ant produces T^k shortest path and two cross-sites are selected at random and the cities in between the cross-sites are inverted (Fig.3).

3	6	2	5	1	7	4
3	6	1	5	2	7	4

Fig.3. Inversion of bits

All the bits are compressed and the bits between the two cross-sites are written at the end (Fig.4).

Figure.4.

3	6	2	5	1	7	4
3	6	7	4	2	5	1

Fig.4. Compression of Bits

Once the shortest path is obtained by the above three methods, each city is chosen and positioned at different locations without changing the order of the other cities and the shortest path is found.

Ants Net Algorithm

Step-1:

Initialize

Set $t := 0$ (t is the time counter)

Set $NC := 0$ (NC is the cycle counter)

For every edge (i,j) set an initial value of

$\tau_{ij} = C$ a random value for trail intensity and $\Delta\tau_{ij} = 0$

Place m ants at n nodes (in our problems $m = n$)

Step-2:

Set $s := 1$ {s is the tabu list index}

For $k := 1$ to m do

Place the starting town of k_{th} ant in $tabu_k(s)$

Ants Net Algorithm Contd…

Step-3:

Repeat until tabu list is full {this step will be repeated (n-1) times}

Set s:= s+1

For k:=1 to m do

Choose the town to move to with a probability $p_{ij}^{k}(t)$ given in (6) {at time k_{th} ant is on town i = $tabu_k(s-1)$}

Move k_{th} ant to town j

Insert town j in $tabu_k(s)$}

Step-4:

For k:=1 to m do

Move the k^{th} ant from $tabu_k(n)$ to $tabu_k(1)$}

Compute the length L_k of the tour described by the ant k

Update the shortest tour found

For every edge (i,j)

For k:=i to m do

$\Delta\tau_{ij} = Q/L_k$ if (i tour described by $tabu_k$ (normally Q is taken as 1)

0 otherwise

Ants Net Algorithm Contd…

Step-5:

For every edge (i,j)
According to (8)
Set $t:=t+n$
Set $NC:=NC+1$
For every edge (i,j) set 0

Step-6:

If ($NC < NC_{max}$) and (not stagnation behavior)
Then
    Empty all tabu lists
    Go to Step 2
Else
    Print shortest tour

Ants Net Algorithm Contd…

Step-7:

Once shortest tour from ACA is obtained perform inversion or compression by GA

Step-8:

Once the shortest path is obtained by the above three methods, each city is chosen and positioned at different locations without changing the order of the other cities and the shortest path is found

Numerical Examples

Examples

Problem – 1 Un symmetric TSP ($d_{ij} \neq d_{ji}$)

Five cities 1,2,3,4,5 and the distances between are given in table.

Answer 15 1-2-3-4-5-1 (un symmetric tsp)

Cities	1	2	3	4	5
1	0	2	5	7	1
2	6	0	3	8	2
3	8	7	0	4	7
4	12	4	6	0	5
5	1	3	2	8	0

Examples Contd…

Step 1

At $t = 0$, $NC = 0$, $\tau_{ij} = C = 0.2$ a random number is assumed and the τ_{ij} matrix is given by

$$\tau_{ij} = \begin{bmatrix} - & .2 & .2 & .2 & .2 \\ .2 & - & .2 & .2 & .2 \\ .2 & .2 & - & .2 & .2 \\ .2 & .2 & .2 & - & .2 \\ .2 & .2 & .2 & .2 & - \end{bmatrix} \text{ and } \Delta\tau_{ij} = [0]$$

$$\eta_{ij} = \begin{bmatrix} - & 0.5 & 0.2 & 0.144 & 1 \\ 0.166 & - & 0.333 & 0.125 & 0.5 \\ 0.125 & 0.144 & - & 0.25 & 0.144 \\ 0.083 & 0.25 & 0.166 & - & 0.2 \\ 1 & 0.333 & 0.5 & 0.125 & - \end{bmatrix}$$

Examples Contd…

Step 2

$s=1$; $k=1$. Place the ant 1 at node 1 the probabilities of choosing the shortest path by that ant is shown below

i=1	τ_{ij}	η_{ij}	$\tau_{ij}\eta_{ij}^2$	p_{ij}
2	0.2	0,5	0.05	0.19
3	0.2	0.2	0.008	0.03
4	0.2	0.144	0.0041	0.015
5	0.2	1	0.2	0.76

$\sum 0.26214$

since p_{15} is the highest and 1 moves to 5 and tabu list is $\begin{Bmatrix} 1 \\ 5 \end{Bmatrix}$

Examples Contd…

Step 3

Place the ant at 5 and let us calculate the probability matrix as shown below

i=5	τ_{ij}	η_{ij}	$\tau_{ij}\eta_{ij}^2$	p_{ij}
2	0.2	0.33	0.0258	0.2907
3	0.2	0.5	0.05	0.667
4	0.2	0.125	0.0031	0.0417
			Σ 0.0749	

Now p_{53} is the highest and the ant moves from city 5 to 3 and the tabu

list, is $\begin{Bmatrix} 1 \\ 5 \\ 3 \end{Bmatrix}$

Examples Contd…

Place the ant at 3 and calculate the probabilities to the paths and the matrix is shown below

i=3	τ_{ij}	η_{ij}	$\tau_{ij}\eta_{ij}^2$	p_{ij}
2	0.2	0.144	0.0044	0.249
4	0.2	0.25	0.125	0.75
			Σ 0.0166	

Now p_{34} is the highest and the ant at 3 moves to 2 and the tabu list is

$$\begin{Bmatrix} 1 \\ 5 \\ 3 \\ 4 \end{Bmatrix}$$

Place the ant at 4 and it can go to only city 2 and the path is 1 - 5 - 3 - 4 - 2 - 1 and the distance of the path taken by the ant is

1 + 2 + 4 + 4 + 6 = 17

Examples Contd…

Step 4

$\Delta\tau_{ij}$ is calculated from (7) as 1/17=0.06

$$\Delta\tau_{ij}^{1} = \begin{bmatrix} - & - & - & - & 0.06 \\ 0.06 & - & - & - & - \\ - & - & - & 0.06 & - \\ - & 0.06 & - & - & - \\ - & - & 0.06 & - & - \end{bmatrix}$$

$\Delta\tau_{ij}$ similarly for other ants the distance of the path traveled is calculated and computed as shown above for all the ants.

Examples Contd…

Step 5

Now is updated using (8) as

$$\tau_{ij}^{(1)} = (1-0.5)\tau_{ij}(0) + \Delta\tau_{ij}(1) = 0.5\times 0.2 + 0.06 = 0.16$$

go to step no 2 and repeat.

Assume 5 - 1 - 2 - 3 - 4 - 5 is the shortest path obtained by ant colony optimization and we are ready to apply the inversion operator of GA. Choose the cross sites at random as 2 and 4 and the real values between the sites are inverted as shown below.

5	1	2	3	4	5
5	1	3	2	4	5

And the length is given as 1 + 5 + 7 + 8 +5 = 26

Examples Contd…

If the bits are compressed between the cross sites we get 5 – 1 – 4 – 2 – 3 – 5 and the length of the route 1 + 7 + 4 + 3 + 7 = 22.

The shortest of all the three in the first iteration is 17 and if it goes for many iterations this converges to 15 and the path is given as **5-1-2-3-4-5 and the shortest length is 15.**

Examples Contd…

Problem – 2: answer 179 6-1-4-3-5-2-6 (un symmetric tsp) using ants net 179 3-5-2-6-1-4-3

0	62	21	55	86	85
55	0	41	85	16	28
76	51	0	90	25	11
92	18	15	0	30	3
89	38	87	70	0	65
18	88	41	83	19	0

Examples Contd…

Problem – 3: answer 45 1-7-2-6-4-3-5-1 (un symmetric tsp) using ants net 42 3-5-2-6-4-1-7-3

0	8	14	8	6	10	3
8	0	12	7	6	5	5
10	9	.0	13	5	13	10
7	6	13	0	7	10	8
7	4	9	10	0	6	9
8	5	13	7	6	0	4
4	5	11	9	6	5	0

Examples Contd…

Problem – 4: answer 190   2-3-1-4-7-6-5-2 (symmetric tsp) using ants net

190   2-3-1-4-7-6-5-2

0	35	20	25	40	30	35
35	0	15	50	45	65	70
20	15	0	25	45	50	55
25	50	25	0	65	45	40
40	45	45	65	0	45	40
30	65	50	45	35	0	45
35	70	55	40	45	10	0

Examples Contd…

Problem – 5: answer 73.07 1-7-3-4-2-6-5-1 (symmetric tsp) using ants net 73.07 5-6-7-4-3-7-1-5

0.0	19.24	23.77	23.71	13	17.03	14.14
19.24	0.0	9.85	8.25	6.71	2.83	15.81
23.77	9.85	0.0	17.12	11.66	9.22	13.6
23.71	8.25	17.12	0.0	13.6	10.77	23.7
13	6.71	11.66	13.6	0.0	4.13	10.4
17.03	2.83	9.22	10.77	10.4	0.0	13.04
14.14	15.81	13.6	23.7	10.4	13.04	0.0

Examples Contd…

Problem – 6: answer 33.8 1-9-5-6-4-7-10-2-3-8-1 L (un symmetric tsp)
using ants net 33.8 7-10-2-3-8-1-9-5-6-4-7

0	51	55	90	41	63	77	69	0.1	23
50	0	0.1	64	8	53	0.1	46	73	72
30	77	0	21	25	51	47	16	0.1	60
65	0.1	6	0	2	9	17	5	26	42
0.1	94	0.1	5	0	0.1	41	31	59	48
79	65	0.1	0.1	15	0	17	47	32	43
76	96	48	27	34	0.1	0	0.1	25	0.1
0.1	17	9	27	46	15	84	0	0.1	24
56	7	45	39	0.1	93	67	79	0	38
30	0.1	42	56	49	77	76	49	23	0

Some more realistic problems are selected from web site:

http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95 and shown in **Table-1**

Examples Contd…

TABLE. 1 EXAMPLE PROBLEMS FROM TSPLIB95
(Values given in brackets are given in **TSPLIB95**)

Problem	Shortest distance Obtained by ANTS NET - Path
Burma14(TSP-sy)	30.87 (33.23) - 8 - 13- 7 - 12 - 6 - 5 - 4 - 3 - 14 - 2 - 1 - 10 - 9 - 11 - 8
BR17(TSP - unsy)	40.09 (39) - 2 - 14 - 3 - 12 - 1 - 16 - 15 - 7 - 6 - 5 - 4 - 17 - 9 - 8 - 13 - 11 - 10 - 2
ULYS22(TSP - sy)	77.84 (70.53) - 14 - 12 - 13 - 3 - 2 - 17 - 18 - 4 - 22 - 8 - 1 - 16 - 15 - 5 - 11 - 9 - 10 - 21 - 20 - 19 - 7 - 6 - 14
BAYS29(TSP - sy)	2125 (2020) - 23 - 8 - 24 - 27 - 16 - 19 - 13 - 1 - 28 - 6 - 12 - 9 - 5 - 26 - 29 - 3 - 2 - 21 - 20 - 10 - 4 - 15 - 17 - 18 - 14 - 22 - 11 - 25 - 7 - 23
FTV34(TSP - unsy)	1501 (1286) - 27 - 30 - 26 - 25 - 24 - 28 - 29 - 2 - 17 - 16 - 15 - 5 - 7 - 6 - 31 - 34 - 3 - 4 - 1 - 14 - 13 - 10 - 33 - 8 - 9 - 11 - 12 - 32 - 19 - 20 - 21 - 18 - 22 - 23 - 27
P43(TSP - unsy)	5639.1 (5620) - 36 - 1 - 5 - 38 - 27 - 26 - 25 - 23 - 24 - 22 - 40 - 39 - 43 - 42 - 41 - 18 - 17 - 16 - 21 - 20 - 19 - 15 - 14 - 13 - 35 - 34 - 33 - 32 - 12 - 11 - 10 - 9 - 8 - 7 - 6 - 31 - 30 - 29 - 28 - 9 - 3 - 2 - 37 - 36
Att48(tsp - sy)	36032.7 (10628) 40 - 9 - 1 - 8 - 38 - 31 - 44 - 18 - 7 - 28 - 36 - 30 - 6 - 37 - 19 - 27 - 43 - 17 - 22 - 3 - 22 - 16 - 11 - 41 - 34 - 29 - 2 - 4 - 26 - 35 - 45 - 24 - 10 - 42 - 48 - 5 - 32 - 39 - 25 - 14 - 23 - 13 - 21 - 47 - 20 - 33 - 46 - 15 - 40

Conclusions

- ACA and GA is used to solve symmetric and un symmetric TSP.
- It is a novel and very promising research field.
- It is derivative free optimization technique.
- It does not fall to local minima/maxima.
- Lot of application areas.
- More information on ACA from the website of Dorigo:

  "http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html"

References

1. Dorigo,M. and Gambardalla,L.M., Ant Colonies for the Traveling Salesman Problem, Bio Systems, 43, pp 73-81,1997.
2. Dorigo,M. and Gambardalla,L.M., Ant Colony System: A Cooperative Learning Application to Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation,* Vol.1, No.1, April.1997.
3. Dorigo .M. http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html, 1999.
4. Reinelt,G.,

   http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95, 1994.