Solving Quadratic Assignment Problems (QAP) using Ant Colony System

Ajay S. Bidyarthy*, Vivek Gupta**
09012305*, 09010367**

April 15, 2013

Aim and Motivation

- **Aim:**
  - Solution to Quadratic Assignment Problems (QAP) using Ant Colony System

- **Motivation**
  - The quadratic assignment problem (QAP) is one of the most difficult problems in NP-hard class.
  - It has applications in several areas such as:
    - Opearional Research
    - Parallel and Distributed Computing and
    - Combinatorial data Analysis
  - Some famous combinatorial optimization problems such as TSP, maximal clique, isomorphism and graph partitioning can be formulated as a QAP.
  - In this project, we use a well known heuristic algorithm, the ant colony system, to solve some of the real applicable QAP.

Survey for the quadratic assignment problem in the existing literature

- The QAP was introduced by Koopmans and Beckmann in 1957 as a mathematical model for the location of indivisible economical activities
- In 1972, Dickey and Hopkins used QAP for assignment of buildings in University campus.
- In 1974, Francis and White used QAP to develop a decision framework for assigning new facilities.
- In 1976, Geoffrion and Graves used QAP for solving scheduling problems.
- In 1978, krarup and Pruzan used QAP in archeology.
- In 1987, Bokhari used QAP in parallel and distrubuted computing.
- In 2003, Rabak and Sichman used QAP in placement of electronic components.

Problem formulation and research work

- Identifying a permutation matrix **X** of dimension $n \times n$ (whose elements X_{ij} are 1 if activity j is assigned to location i and 0 in the other cases) such that:

$$\min z = \sum_{i,j=1}^{n} \sum_{h,k=1}^{n} d_{ih} f_{jk} X_{ij} X_{hk}$$

Subject to

$$\sum_{i=1}^{n} X_{ij} = 1 \quad \text{for } j = 1, 2, 3, ..., n$$
$$\sum_{j=1}^{n} X_{ij} = 1 \quad \text{for } i = 1, 2, 3, ..., n$$
$$X_{ij} \in (0, 1) \quad \text{for } i, j = 1, 2, 3, ..., n \tag{1}$$

Where $d_{ih} \in D$ distance matrix and $f_{jk} \in F$ flow matrix.

Problem formulation and research work

- ACS have been applied to various combinatorial optimization problems.
- ACS is competitve with other nature inspired algorithms such as particle swarm optimization, simulated annealing and evolutionary computation algorithms.
- The running time of ACS is $O(mn^2k)$, which is much better for big graphs than the recursive solution $O(n!)^3$ where n is the number of cities that may be visited by m ants for k cycles.

Problem formulation and research work

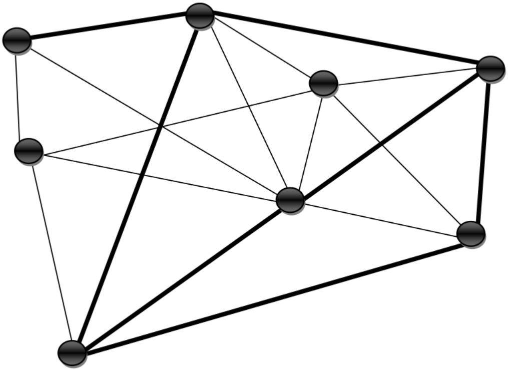
- ACS have been applied to various combinatorial optimization problems.
- ACS is competitve with other nature inspired algorithms such as particle swarm optimization, simulated annealing and evolutionary computation algorithms.
- The running time of ACS is $O(mn^2k)$, which is much better for big graphs than the recursive solution $O(n!)^3$ where n is the number of cities that may be visited by m ants for k cycles.

QAP to SBRP Formulation

- School Bus Routing Problem (SBRP)
- QAP to SBRP
- SBRP objectve
- Implementation assumptions

Ant Colony System Algorithm

- We assume that ants travel from bus stop r to bus stop s.
- Each edge on the graph has length $\delta(r, s)$, and a pheromone measure,$\tau(r, s)$.This pheromone measure is updated every time ant walks over this edge.



: How ants exploit pheromone to find a shortest path

- Transition Rule

$$s = \begin{cases} \arg\max_{u \in J_k(r)} \{[\tau(r,u)]^{\mu}[\eta(r,u)]^{\beta}\} & \text{if } q < q_0 \\ p_k(r,s) & \text{otherwise} \end{cases} \quad (2)$$

Where

- $J_{k(r)}$ is the set of cities that remain to be visited by ant k positioned in city r and $\eta(r,u) = 1/\delta(r,u)$
- $[\mu, \beta]$ are parameters that set the relative importance of pheromone vs. distance
- q_0 is a constant parameter, defined by the range $(0 < q_0 < 1)$, which is selected to establish the exploitation vs. exploration.
- $q =$ is a random number defined by the range $(0 < q < 1)$

Ant Colony System Algorithm continue...

- $$p_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^{\mu}[\eta(r,s)]^{\beta}}{\sum_{u \in J_k(r)}[\tau(r,u)]^{\mu}[\eta(r,u)]^{\beta}} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

  Where
- $P_k(r,s)$ is the chance of city s to be chosen by ant k positioned in bus stop r
- Since the pheromone on the edge is multiplied by η, which depends on δ, better odds are given to shorter edges with more pheromone.

Ant Colony System Algorithm continue...

- After each ant relocation step, the pheromone on all edges is updated according to a local pheromone-updating rule given by:

$$\tau(r,s) = ((1-\rho) \times \tau(r,s) + \rho \times \Delta\tau(r,s)) \tag{4}$$

Where

- ρ is a parameter defined in the range $(0 < \rho < 1)$, describes the local evaporation of pheromone
- $\Delta\tau(r,s)$ is the sum of all pheromone left by ants that used this edge in their last step
- Normally ρ is fixed during the path search. An ant using edge (r,s), normally leaves $1/\delta(r,s)$ pheromone on the edge

- Once all ants have completed a tour, a global pheromone-updating rule is applied. The global updating rule is described by the equation:

$$\tau(r,s) = ((1-\alpha) \times \tau(r,s) + \alpha \times \Delta\tau(r,s) \quad (5)$$

Where

- α is the global evaporation parameter defined by the range $(0 < \alpha < 1)$.

$$\Delta\tau(r,s) = \begin{cases} \frac{1}{\text{length of global best tour}} & \text{If } (r,s) \text{ belongs to this global be} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

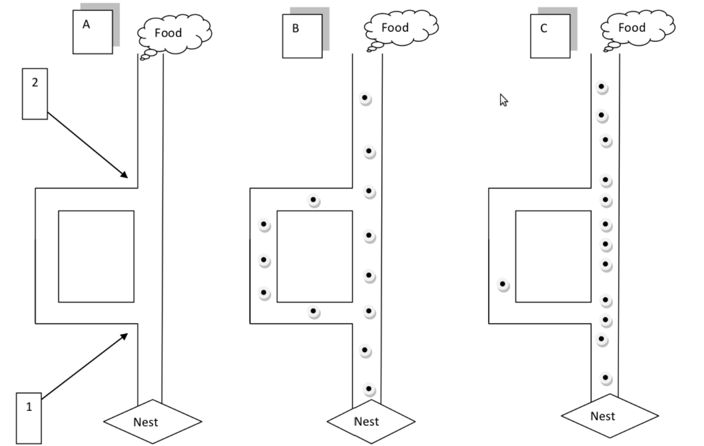
ACO for SBRP Implementation

- Show code
- Dummy input data
- output

Benchmark Algorithm and Problems

- Algorithms
  - Genetic Algorithm
  - Particle Swarm Optimization
  - Evolutionary Algorithm
  - Simulated Annealing
- Problems
  - Travelling Salesman Problem
  - Vehicle Routing problems
  - Time tabling problem
  - Campus designing problem
  - Scheduling problems
  - Elevator Problem

How Ants Find a Shortest Path



: How ants exploit pheromone to find a shortest path

Running time complexity

Assuming that n is the number of cities that may be visited by m ants for k cycles.

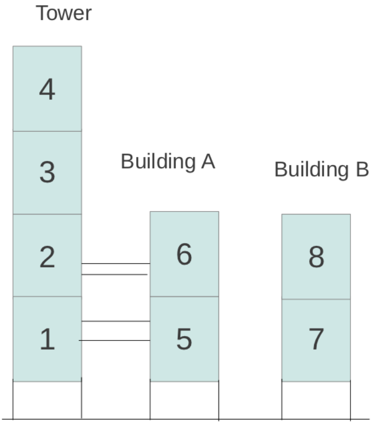
Every ants has to decide among n cities, using probability function, this requires n calculations. Thus for all ants combined, one step requires $m \times n$ calculations.

To complete 1 cycle, one ant must go through all n cities. This requires $m \times n \times n$ calculations for each cycle.

Therefore the running time is $O(mn^2k)$, which is much better for big graphs than the recursive solution - $O(n!)^3$.

ACO for QAP

Consider the QAP of order 8



: Position of the units in the three buildings available

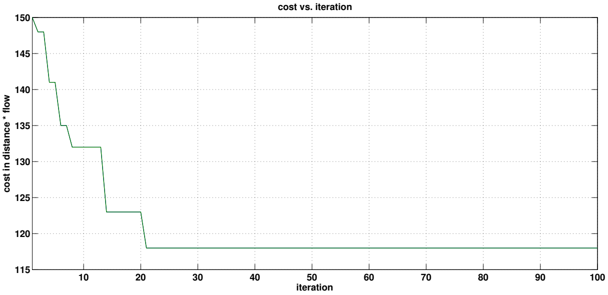
Distance Flow Matrix

$$DF = \begin{bmatrix} 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 \\ 5 & 0 & 1 & 2 & 2 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 & 3 & 2 & 1 & 2 \\ 4 & 0 & 0 & 0 & 4 & 3 & 2 & 1 \\ 1 & 2 & 0 & 5 & 0 & 1 & 2 & 3 \\ 0 & 2 & 0 & 2 & 10 & 0 & 1 & 2 \\ 0 & 2 & 0 & 2 & 0 & 5 & 0 & 1 \\ 6 & 0 & 5 & 10 & 0 & 1 & 10 & 0 \end{bmatrix}$$

For ants $= 5$, $\mu = -1$ (weight of pheromone), $\beta = 1$ (weight of heuristic info), $\rho = 0.9$ (evaporation parameter), $Q = 10$ (Constatnt for pheromone update).

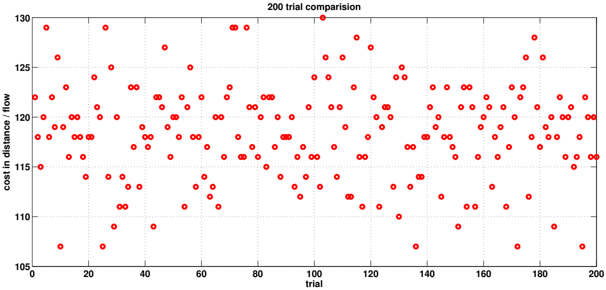
Iteration to Solution

Figure 4 shows a ACO solution for the QAP input matrix shown above, this is a sub optimal solution at 118 whereas the ideal solution is 107.



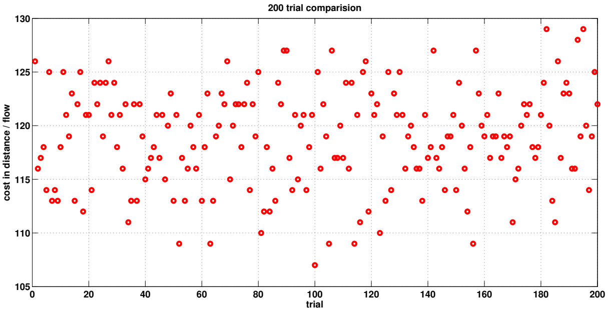
: Typical Iteration to Solution

Demonstration of local and global optimization



: Comparison with the number of ants = number of nodes (8)

Demonstration of local and global optimization



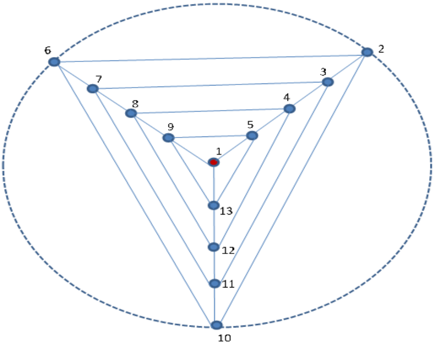
: Comparison with number of ants fixed at four

Results for QAP

Name	n	given opt	our opt	time	max iter
chr12a	12	9552	10633	5.1224	2000
chr12c	12	11156	10119	5.1951	2000
esc16b	16	292	150	0.0736	2000
esc16d	16	16	31	0.0312	2000
nug15	15	1150	667	6.9119	2000
scr12	12	31410	19111	5.0277	2000
scr15	15	51140	35737	6.9073	2000
chr15a	15	9896	13562	6.8983	2000

Table: Results for standard QAP problems

Radial input



: Bus Stops are aligned radially outwards from the school on three radials

Radial input

school_id(down)/bus_stop_id(right)	2	3	4	5	6	7	8	9	10	11	12	13
1	4	4	4	4	4	4	4	4	4	4	4	4

Bus stop Vs. students matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	10	7.5	5	2.5	10	7.5	5	2.5	10	7.5	5	2.5
2	10	0	2.5	5	7.5	10	10	10	10	10	10	10	10
3	7.5	2.5	0	2.5	5	10	7.5	7.5	7.5	10	7.5	7.5	7.5
4	5	5	2.5	0	2.5	10	7.5	5	5	10	7.5	5	5
5	2.5	7.5	5	2.5	0	10	7.5	5	2.5	10	7.5	5	2.5
6	10	10	10	10	10	0	2.5	5	7.5	10	10	10	10
7	7.5	10	7.5	7.5	7.5	2.5	0	2.5	5	10	7.5	7.5	7.5
8	5	10	7.5	5	5	5	2.5	0	2.5	10	7.5	5	5
9	2.5	10	7.5	5	2.5	7.5	5	2.5	0	10	7.5	5	2.5
10	10	10	10	10	10	10	10	10	10	0	2.5	5	7.5
11	7.5	10	7.5	7.5	7.5	10	7.5	7.5	7.5	2.5	0	2.5	5
12	5	10	7.5	5	5	10	7.5	5	5	5	2.5	0	2.5
13	2.5	10	7.5	5	2.5	10	7.5	5	2.5	7.5	5	2.5	0

Distance Matrix of Bus Stops

: Bus stops Vs. students matrix and distance matrix of bus stops (radial)

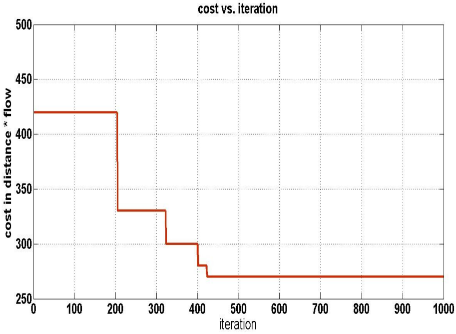
Radial output

Output for bus 1

Cheapest Cost for bus id 1 is 270. Bus id 1 is assigned to bus stops in following order: $->5->2->6->10->9->4->13->3->7->12->8->11->1$. CPU time taken (in Seconds): 1.8778.

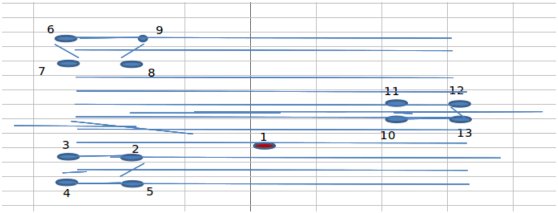
Radial output

Figure 9 represents cost in distance multiplied by flow Vs. iterations.



: Cost in distance multiplied by flow Vs. iterations

Clustered input



: Bus Stops are clustered in 4 clusters around the school in all directions

Clustered input

school_id(down)/bus_stop_id(right)	2	3	4	5	6	7	8	9	10	11	12	13
1	4	4	4	4	4	4	4	4	4	4	4	4

Bus stop Vs. students matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	3	4	6	5	11	9	8	10	4	5	6	5
2	3	0	1	3	2	10	8	7	9	7	8	9	8
3	4	1	0	2	3	9	7	8	10	8	9	10	9
4	6	3	2	0	1	11	9	10	12	10	11	12	11
5	5	2	3	1	0	12	10	9	11	9	10	11	10
6	11	10	9	11	12	0	2	3	1	11	10	11	12
7	9	8	7	9	10	2	0	1	3	9	8	9	10
8	8	7	8	10	9	3	1	0	2	9	8	9	10
9	10	9	10	12	11	1	3	2	0	11	10	11	12
10	4	7	8	10	9	11	9	9	11	0	1	2	1
11	5	8	9	11	10	10	8	8	10	1	0	1	2
12	6	9	10	12	11	11	9	9	11	2	1	0	1
13	5	8	9	11	10	12	10	10	12	1	2	1	0

Distance Matrix of Bus Stops

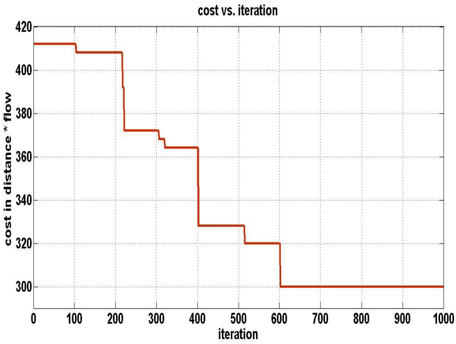
: Bus stops Vs. students matrix and distance matrix of bus stops (clustered)

Clustered output

Output for bus 1

Cheapest Cost for bus id 1 is 300. Bus id 1 is assigned to bus stops in following order: $->2->6->9->4->5->12->13->11->10->7->8->3->1$. CPU time taken (in Seconds): 1.9089.

clustered output
Figure 12 represents cost in distance multiplied by flow Vs. iterations.



: Cost in distance multiplied by flow Vs. iterations

Summary and Future Work

- Summary
  - Successfully applied the Ant colony System to the Quadratic Assignment Problem (QAP) and School Bus Routing Problem (SBRP).
  - The results obtained using ACO are better than the results obtained using standard QAP approach in some cases.
  - For SBRP, three test cases were considered i.e. radial , clustered and random arrangement of bus stops.
  - For radial arrangement of bus stops, the optimal route obtained are radial, whereas for clustered arrangement of bus stops, the optimal routes are clustered.
- Future Work
  - As future work ACO can be applied to distributed environment to further increase the efficiency of the algorithm.

Koopmans, T.C. and Beckmann, M.J. (1957). Assignment problems and the location of economic activities. *Econometrica*, 25, 53-76.

Francis, R.L. and White, J.A. (1974). Facility layout and location: An analytical approach. Prentice-Hall, Englewood Cliffs, New Jersey

Krarup, J. and Pruzan, P.M. (1978). Computer-aided layout design. *Mathematical Programming Study*, 9, 75-94.

Hubert, L. and Schulz, J. (1976). Quadratic assignment as a general data analysis strategy.*British Journal of Mathematical Psychology*, 29, 190-241.

Dickey, J.W. and Hopkins, J.W. (1972). Campus building arrangement using topaz. *Transportation Research*, 6, 59-68.

Rabak, C.S., Sichman, J.S. (2003). Using A-Teams to optimize automatic insertion of electronic components, *Advanced Engineering Informatics*, 17. (2), 95-106.