Metaheuristic Optimization: Ant Colony Optimization

Adaptive and Cooperative Algorithms (ECE 457A)

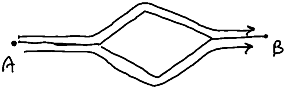
ECE, MME, and MSCI Departments,
University of Waterloo, ON, Canada

Course Instructor: Benyamin Ghojogh

Fall 2023

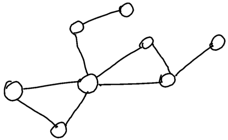
Introduction

- **Ant Colony Optimization (ACO)** is a family of metaheuristic optimization algorithms which are mostly useful for **optimization on graphs**.
- Ant colony is a **swarm optimization** method which is also **probabilistic**.
- The original ant colony is also named **Simple Ant Colony Optimization (SACO)**.
- One of the initial ant colony algorithms is named the **Ant System**, proposed in 1992 in Marco Dorigo's PhD thesis [1]. Another initial ant colony algorithm was by Luca Gambardella in 1997 for the traveling salesman problem [2].
- Two of important scholars in the area of the ant colony algorithms:
  - **Marco Dorigo**: `https://scholar.google.com/citations?user=PwYT6EMAAAAJ&hl=en&oi=sra`
  - **Luca Maria Gambardella**: `https://scholar.google.com/citations?user=SCCwW3YAAAAJ&hl=en&oi=sra`
- Consider two paths (1) and (2) on the ground from point A to point B. Initially a colony of ants pass from the two paths randomly but they **gradually select one of the paths and pass from on them only thereafter**.
- This is because of the **pheromone** which every ant puts behind itself on its traversed path. The ants tend to move on the paths having more pheromone. In this way, ants can help each other in navigation.



Graph

- A graph G is represented by $G = (\mathcal{V}, \mathcal{E})$
  where $\mathcal{V}$ is the set of vertices (nodes) and $\mathcal{E}$ is the set of edges (links) between the vertices.



- There may or may not be an edge or edges between any two vertices. If there is an edge between every pair of vertices, the graph is called a **complete graph** or a **fully-connected graph**.
- The edges may or may not be **directed**. The former is a directed graph and the latter is an underacted graph.
- The edges of graph may or may not have weights. If they have weights, the graph is called a **weighted graph**. A unweighted graph can be considered as a special case of weighted graph with equal weights.
- A **path** between two vertices on a graph is a set of edges where by passing them we can reach from one vertex to another vertex. If the graph is directed, then the path is also **directed** and the direction of its edges should be considered.

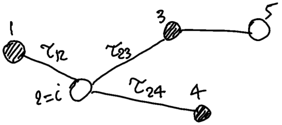
Selecting Edges for the Ant's Path

Probability of Selecting Edges

- Every edge in the graph has some cost or fitness as its weight. Let c_{ij} denote the cost of the edge between vertices i and j.
- Every edge in the graph has some **pheromone** on it. Let $\tau_{ij}(t) \geq 0$ denote the pheromone of the edge between vertices i and j at iteration t. For undirected graphs, we have $\tau_{ij}(t) = \tau_{ji}(t)$. We can initialize the pheromones of all edges to equal amount or small positive random values at the start of ACO algorithm.
- In ACO, we have a colony of n ants. At every iteration, we put every ant on the **starting vertex**.
- When the k-th ant is on the vertex i, the **probability** that it goes from vertex i to vertex j is calculated as:

$$P_{ij}^{(k)} = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)}{\sum_{l \in \mathcal{N}_i} \tau_{il}^{\alpha}(t)} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{if } j \notin \mathcal{N}_i, \end{cases} \tag{1}$$

  where $\mathcal{N}_i$ is the set of neighbor vertices for the vertex i and $\alpha > 0$ is a scalar hyperparameter.
- Therefore, it is **only possible to go to neighbor vertices**. Also, if the edge has **more pheromone**, the **more probable** it is for the ant to choose that edge for its path.



Probability of Selecting Edges

- It might be better to add some heuristic information to the formulation of the probability.
- For example, the **probability** that the ant goes from vertex i to vertex j can be:

$$P_{ij}^{(k)} = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)\, \eta_{ij}^{\beta}(t)}{\sum_{l \in \mathcal{N}_i} \tau_{il}^{\alpha}(t)\, \eta_{il}^{\beta}(t)} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{if } j \notin \mathcal{N}_i. \end{cases} \quad (2)$$

  where $\alpha > 0$ and $\beta > 0$ are scalar hyperparameters.
- The $\eta_{ij}(t)$ can be any value whose larger value is more desired. Usually, we use a fixed value not dependant on the iteration index. For example, it can be:

$$\eta_{ij}(t) = \eta_{ij} = \frac{1}{c_{ij}}, \quad (3)$$

  where c_{ij} denote the cost of the edge between vertices i and j.
- This probability is used in the **Ant System** algorithm (1996) [3].
- In some applications such as embedded systems, calculation of power is difficult and we use the following instead:

$$P_{ij}^{(k)} = \begin{cases} \frac{\alpha\tau_{ij}(t)+(1-\alpha)\eta_{ij}(t)}{\sum_{l \in \mathcal{N}_i} \alpha\tau_{il}(t)+(1-\alpha)\eta_{il}(t)} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{if } j \notin \mathcal{N}_i, \end{cases} \quad (4)$$

  where $\alpha \in [0, 1]$.

Sampling the Edges

- Assume the k-th ant is on the vertex i and we calculated the probability $P_{ij}^{(k)}$ for all $j \in \mathcal{N}_i$. We can select the next vertex to go to by any of the following methods:
  - **Simple Ant Colony**:

$$\ell \sim P_{ij}^{(k)}, \tag{5}$$

    where $\sim P_{ij}^{(k)}$ is sampling from the discrete probabilities $P_{ij}^{(k)}, \forall j$ using the **roulette wheel technique** (see the lecture of genetic algorithm).
  - **Ant Colony System** (1992-1997) [1, 2]:

$$\ell := \begin{cases} \arg\max_{j \in \mathcal{N}_i} P_{ij}^{(k)} & \text{if } u \leq r \\ \sim P_{ij}^{(k)} & \text{if } u > r, \end{cases} \tag{6}$$

    where ℓ is the next vertex to move the ant to, $u \sim U(0,1)$ is a uniform random number, and $r \in (0,1)$ is a scalar threshold.
    - In this method, we choose the most probable edge with a probability r or sample the edge from the probabilities with a probability $1 - r$.
    - The larger the r is, the more probable is to choose the edge with highest probability, so the more **exploitation** we have. Likewise, the smaller the r is, the more **exploration** we have. Therefore, we can start with larger r in the initial iterations and we decrease r gradually.

Sampling the Edges

- We can have other variants too such as:
  - **Maximum probability and random sampling**:

$$\ell := \begin{cases} \arg\max_{j \in \mathcal{N}_i} P_{ij}^{(k)} & \text{if } u \leq r \\ \sim \mathcal{N}_i & \text{if } u > r, \end{cases} \tag{7}$$

    where $\sim \mathcal{N}_i$ is random sampling from one of the neighbor vertecies of the vertex i.
  - **Probabilistic sampling and random sampling**:

$$\ell := \begin{cases} \sim P_{ij}^{(k)} & \text{if } u \leq r \\ \sim \mathcal{N}_i & \text{if } u > r. \end{cases} \tag{8}$$

  - **Random sampling**:

$$\ell \sim \mathcal{N}_i. \tag{9}$$

Updating Pheromones

Evaporation of Pheromones

- In the real world, the pheromones gradually evaporate.
- Likewise, we decrement (evaporate) the pheromones gradually with the iterations:

$$\tau_{ij}(t+1) := (1-\rho)\tau_{ij}(t), \tag{10}$$

  where $\rho \in (0,1)$ is a scalar hyperparameter.
- If we do not evaporate the pheromones, the ACO algorithms usually do not converge especially in the problems with a large number of vertices.

Updating Pheromones by Ants

- In the real world, the pheromones are updated (strengthened) by ants moving on them.
- Likewise, we update (strengthen) the pheromones gradually with the iterations:

$$\tau_{ij}(t+1) := \tau_{ij}(t) + \Delta\tau^{(k)}(t), \quad (11)$$

  where $\Delta\tau^{(k)}(t) \geq 0$ is the contribution of the k-th ant in updating the pheromone for the edge (i,j). This update is performed for all edges in the path traversed by the k-th ant.
- The update $\Delta\tau^{(k)}(t) \geq 0$ can be calculated in various ways such as:

$$\Delta\tau^{(k)}(t) := \frac{1}{f^{(k)}}, \quad \text{or}, \quad (12)$$

$$\Delta\tau^{(k)}(t) := \frac{1}{n^{(k)}}, \quad (13)$$

  where $f^{(k)}$ is the total cost of the path of the k-th ant, i.e., the summation of the costs of the edges which the ant has passed. The $n^{(k)}$ is the $n^{(k)}$ is the number of edges which the k-th ant has passed.
- The above equations make sense because:
  - The paths with lowest cost should be strengthened.
  - The shortest paths are usually suitable (based on the application) and should be strengthened.

Updating Pheromones by Ants

- If the edges have fitness values which should be maximized, we can alternatively use:

$$\Delta\tau^{(k)}(t) := f^{(k)}. \tag{14}$$

- We can either evaporate the pheromones and then update them by the ants or we can combine these two steps together:

$$\tau_{ij}(t+1) := (1-\rho)\tau_{ij}(t) + \rho\Delta\tau^{(k)}(t), \tag{15}$$

where $\rho \in (0, 1)$ is a scalar hyperparameter.

Updating Pheromones by Ants

- Let $\mathcal{P}^{(k)}$ denote the set of edges in the path traversed by the k-th ant. There are various variants for updating the pheromones. Some of them are:
  - **Ant Cycle** [1]:

$$\Delta\tau^{(k)}(t) := \begin{cases} \frac{\gamma}{f^{(k)}} & \text{if } (i,j) \in \mathcal{P}^{(k)} \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

    where $\gamma > 0$ is a scalar hyperparameter. This update is the Eq. (12). This update uses the cost of all edges in the path for updating the pheromone of every edge in the path.
  - **Ant Quantity** [1]:

$$\Delta\tau^{(k)}(t) := \begin{cases} \frac{\gamma}{c_{ij}} & \text{if } (i,j) \in \mathcal{P}^{(k)} \\ 0 & \text{otherwise,} \end{cases} \quad (17)$$

    where c_{ij} is the cost of the edge (i,j). This update uses the cost of every edge in the path for updating the pheromone of that edge itself.
  - **Ant Density** [1]:

$$\Delta\tau^{(k)}(t) := \begin{cases} \gamma & \text{if } (i,j) \in \mathcal{P}^{(k)} \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

- Experiments have shown that ant cycle works best and ant density works worst.

Some Variants of Ant Colony Optimization

Some Variants of Ant Colony Optimization

- There are various variants of ant colony optimization. We review some of the variants here.
- We can only update the pheromones of the path traversed by the **best ant** having least cost. However, this only has **exploitation** and it leads to getting **stuck in the local best** very soon.

Some Variants of Ant Colony Optimization

- **Ant Colony System** (1992-1997) [1, 2]:
  - It uses Eq. (6) for edge sampling.
  - For **exploitation**, it updates the pheromones of the best ant having least cost in its path:

$$\tau_{ij}(t+1) := (1-\rho_1)\tau_{ij}(t) + \rho_1 \Delta\tau^{(k)}(t). \tag{19}$$

  - Moreover, it does not ignore updating the other ants in order to have **exploration**. It updates the pheromones of the other ants by a constant value:

$$\tau_{ij}(t+1) := (1-\rho_2)\tau_{ij}(t) + \rho_2 \frac{1}{|\mathcal{V}| \times f^*}, \tag{20}$$

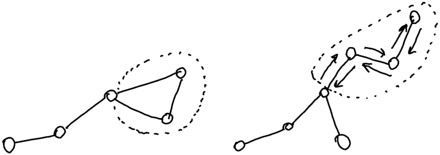
    where $\rho_1, \rho_2 > 0$ are the scalar hyperparameters, $|\mathcal{V}|$ is the number of vertices in the graph, and f^* is an estimate of the best (least) cost value in the problem.
  - It is possible to update multiple best ants and not only the top best ant using Eq. (19).
  - There are variants of ant colony system for defining the best ant(s):
    - **best global ant**: The best ant can be the ant which have seen the least cost so far from the start of algorithm.
    - **best local ant**: The best ant can be the ant which have seen the least cost in the current iteration.
    - In the successive iterations, we can alternate between choosing the best local ant and the best global ant as the best ant.

Some Variants of Ant Colony Optimization

- **Max-Min Ant System** (2000) [4]:
  - The pheromones are only in a range $[\tau_{\text{min}}, \tau_{\text{max}}]$.
  - In the initial iteration, the pheromones of all edges are initialized to τ_{max}. This is to give all paths a chance of being selected.
  - The pheromones of all edges evaporate gradually. The pheromones of the best ants get updated. As a result, the pheromones of the edges remain where the best ants have passed from.
  - It alternates between choosing the best local ant and the best global ant as the best ant. If gradually many of the ants pass from the same path, it updates the best local ant for several iterations. If still many of the ants pass from that same path, it resets all the pheromones to τ_{max} and redoes the algorithm. However, it always saves the best solutions and returns the best of best solutions finally.

Some Variants of Ant Colony Optimization

- **Ant Tabu (ANTabu)** (1999) [5]:
  - If an ant goes to some part of the graph but finds out that the costs are large on the edges of that part of the graph, it can put those edges in a Tabu list not to repeat that path again in the next iterations.
  - Moreover, if the ant find out that there is a loop at some part of the graph, it can put the edges of that loop in the Tabu list.



  - The Tabu list can be either used for that individual ant ot all ants in the next iterations.
  - Ant Tabu (ANTabu) [5] is a combination of the ant colony optimization and Tabu search [6, 7].
  - It updates the pheromones as:

$$\tau_{ij}(t+1) := (1-\rho)\tau_{ij}(t) + \rho(\frac{1}{f^{(k)}})(\frac{f_{\max} - f^{(k)}}{f_{\min}}), \quad (21)$$

where $f_{\min}$ and $f_{\max}$ are the best (least) and worst (largest) costs found so far and $f^{(k)}$ is the cost of the edges traversed by the k-th ant in this iteration.

Some Variants of Ant Colony Optimization

- **Rank-based Ant Colony** (2005) [8]:
  - If the cost values differ significantly, Eq. (21) does not work well enough.
  - Therefore, rank-based ant colony [8] ranks the ants from best to worst in every iteration. Suppose the rank indices for the best and worst ants are zero and $n-1$, respectively. Let the rank of the k-th ant be denoted by $\sigma^{(k)} \in \{0, 1, \dots, n-1\}$.
  - It updates the pheromones as:

$$\tau_{ij}(t+1) := (1-\rho)\tau_{ij}(t) + \rho\Big(\frac{\gamma(n-\sigma^{(k)})}{f^{(k)}}\Big), \tag{22}$$

    where $\gamma > 0$ is a scalar hyperparameter.
  - As expected, the best ants have more contribution in updating the pheromones in this method.

Algorithm of Ant Colony Optimization

Ant Colony Optimization

Algorithm Ant Colony Optimization (ACO)

Initialize $\tau_{ij}(0), \forall i, j \leftarrow U(0, 1)$. Initialize $t = 0$.

while *not converged* **do**
- Place all ants at the source vertex.
- **for** *each ant* $k \in \{1, \ldots, n\}$ **do**
  - $\mathcal{P}^{(k)} = \varnothing$
  - **while** *not reached to destination* **do**
    - $(i, j) \leftarrow$ Select the next edge to traverse by $P_{ij}^{(k)}$
    - $\mathcal{P}^{(k)} \leftarrow \mathcal{P}^{(k)} \cup (i, j)$
  - Remove loops from $\mathcal{P}^{(k)}$ (optional)
  - Calculate $f^{(k)}$
  - **if** $f^{(k)} <$ *best cost* **then**
    - Update the best cost by $f^{(k)}$ and the best path by $\mathcal{P}^{(k)}$
- **for** *each edge* $(i, j) \in \mathcal{E}$ **do**
  - $\tau_{ij}(t + 1) := (1 - \rho)\tau_{ij}(t)$
- **for** *each ant* $k \in \{1, \ldots, n\}$ **do**
  - **for** *each edge* $(i, j) \in \mathcal{E}$ **do**
    - $\tau_{ij}(t + 1) := \tau_{ij}(t) + \Delta\tau^{(k)}(t)$
- $t := t + 1$

Return the path $\mathcal{P}^{(k)}$ of the best solution

Acknowledgment

- Some slides of this slide deck are inspired by teachings of Prof. Saeed Sharifian at the Amirkabir University of Technology, Department of Electrical Engineering.
- Some surveys on ant colony optimization: [9, 10, 11, 12, 13, 14]

References

[1] M. Dorigo, "Optimization, learning and natural algorithms," *Ph.D. Thesis, Politecnico di Milano*, 1992.

[2] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[4] T. Stützle and H. H. Hoos, "Max–min ant system," *Future generation computer systems*, vol. 16, no. 8, pp. 889–914, 2000.

[5] E.-G. Talbi, O. Roux, C. Fonlupt, and D. Robillard, "Parallel ant colonies for combinatorial optimization problems," in *Parallel and Distributed Processing: 11th IPPS/SPDP'99 Workshops*, pp. 239–247, Springer, 1999.

[6] F. Glover, "Tabu search–part I," *ORSA Journal on computing*, vol. 1, no. 3, pp. 190–206, 1989.

[7] F. Glover, "Tabu search–part II," *ORSA Journal on computing*, vol. 2, no. 1, pp. 4–32, 1990.

[8] J.-l. Liu, "Rank-based ant colony optimization applied to dynamic traveling salesman problems," *Engineering Optimization*, vol. 37, no. 8, pp. 831–847, 2005.

References (cont.)

[9] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical computer science*, vol. 344, no. 2-3, pp. 243–278, 2005.

[10] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life reviews*, vol. 2, no. 4, pp. 353–373, 2005.

[11] B. C. Mohan and R. Baskaran, "A survey: Ant colony optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4618–4627, 2012.

[12] S. Katiyar, N. Ibraheem, and A. Q. Ansari, "Ant colony optimization: a tutorial review," in *National conference on advances in power and control*, pp. 99–110, 2015.

[13] A. Akhtar, "Evolution of ant colony optimization algorithm–a brief literature review," *arXiv preprint arXiv:1908.08007*, 2019.

[14] M. Dorigo and T. Stützle, *Ant colony optimization: overview and recent advances.* Springer, 2019.