Ant Colony Optimization

Feng-Cheng Yang

computer automation

Enthusiasm • Leadership • Innovation • Teamwork • Excellence

NTU 國立臺灣大學工業工程學研究所

Ant Colony Optimization

- Introduction
- ACO Development History
- ACO Algorithm
- ACO for TSPs
- ACO for BPPs

Ant Colony Optimization (ACO) Algorithm

- Ant System (AS) was developed by Dorigo in 1990s (formally published in1996)
  - AS is the first ACO algorithm
  - Applied to the traveling salesman problem (TSP)
  - Inspired by the food-searching behavior of real ants
- A population-based heuristic method
  - For optimization problems

The ant colony optimization method

- Successfully applied to solve various combinatorial optimization problems
  - But in general not for continuous problems
    - A few research work on this issue
- A constructive heuristic algorithm
  - Solutions are constructed step by step
  - By a population of agents—artificial ants

Food Search Behaviors of Ants

- Ants communicate with each other by depositing pheromone on the trails
  - Pheromone is subject to evaporation when time goes by
- A colony of ants will travel on the shortest path between the food source and their formicary
  - The Intensity of pheromone on the shorter path is stronger than the lengthy one

ACO algorithm

- **Three mechanisms are used for constructing solutions**
  - **A proportional probabilistic selection mechanism**
    - In object selection process of solution construction
  - **A global and local memory-recording mechanism**
    - Each ant records the process states of solution construction
    - Each ant can refer to a global memory for solution guidance
  - **A pheromone map and an update mechanism**
    - Maintains a pheromone map to guide the ants' solution search
    - Update the map with the obtained knowledge from the solution quest

Ant Colony Optimization Techniques

- Ant System, AS, is the first ACO system
- AS presented an optimal-solution search algorithm
  - Based on indirect pheromone communication between a colony of artificial ants
  - A pheromone map (or matrix) is designed for the solution search guidance
- An artificial ant constructs a solution step by step in each iteration
- The best solution so far evolved iteratively

Other Ant Optimization Systems

- Extended, enhanced, and improved ant systems
  - Ant Colony System (ACS)
  - Rank-Based Version of the Ant System (AS_rank)
  - Max-Min Ant System (MMAS)
- Improvement systems focus on
  - Better selection mechanisms
  - Different pheromone update methods
  - Effectiveness enhancements

ACO for Combinatorial Optimization Problems

- Problems involve a set of “objects”
- ACO solution construction deals with these objects one by one
- The links between two consecutively selected objects reveal optimizing information
  - Object in a TSP→ cities to be visited
    - An artificial ant composes a sequence of these objects (cities) one by one to form a traveling route
  - Object in a BPP→ parcels or items to be packed into capacity-constrained bins
    - An artificial ant repetitively selects an object (a parcel or an item) and a bin to place it, until no object is left

Solution Construction

- Initially, an ant faces a set of non-processed objects
- Objects are dealt with to join either an ordered sequence (for a TSP) or a subset of selected objects (for a BPP)
  - Selecting an object stochastically from a candidate set of non-processed, yet constraint-satisfied, objects
- The selected object is appended to the partial solution that is under construction

Solution: A sequence of objects

$$\mathbf{S} = < s_1, s_2, \ldots, s_z > \quad s_i \in \mathcal{O} = \{1, 2, \ldots, z\}$$

Pheromone Map: $z \times z$ square matrix

$$\mathbf{T} = \left[\tau_{ij}\right]_{z\times z} = \begin{bmatrix} \tau_{11} & \tau_{12} & \cdots & \tau_{1z} \\ \tau_{21} & \tau_{22} & \cdots & \tau_{2z} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{z1} & \tau_{z2} & \cdots & \tau_{zz} \end{bmatrix}$$

Solution: A set of object subsets

$$\mathbf{S} = \left\{S_1, S_2, \ldots, S_g\right\} =$$

$$\left\{\left\{\mathbf{s}_{11}, \mathbf{s}_{12}, \ldots, \mathbf{s}_{1|S_1|}\right\}, \ldots, \left\{\mathbf{s}_{g1}, \mathbf{s}_{g2}, \ldots, \mathbf{s}_{g|S_g|}\right\}\right\}, \quad \sum_{k=1}^{g} \left|S_k\right| = z.$$

Pheromone Map: $z \times g$ matrix

$$\mathbf{T} = \left[\tau_{ij}\right]_{z\times g}$$

Object Selection

- One object is selected from the candidate set
  - Following a proportional probabilistic process
- Selection Probability
  - For object j in the candidate set N_i of object i to succeed object i
  - Calculated from the pheromone value and a heuristic value
    - The heuristic value is calculated using a local optimization approach, in order to guide the selection towards optimality
    - Detailed formulations of the probability calculations are problem-dependent, and different ant systems might have their own designs for them

ACO for Object-sequencing Problems

- **Object-selection**
  - **Selection probability computation**
  - **Global information reference**
    - Ants refer to a pheromone map
      - Usually a two-dimensional matrix
    - Constructed from the links between two objects
  - **A pheromone value τ_{ij} is the probability or the tendency that object i is followed by object j in the processing sequence**

ACO for Object Grouping Problems

- Type 1: Object and group selections
  - Group number is known
  - This case uses a $z \times g$ pheromone matrix, where g is the number of bins
- Type 2: Object selection against a current group
  - Group number is to minimize
  - A pheromone value is the probability of the tendency that object j is grouped with objects is that have been grouped in the current group
  - This case uses a $z \times z$ square matrix

Pheromone Value Update

- Ants refer to the map to guide their searches toward an optimal solution (calculating probabilities for candidates)
- Pheromone values are intensified by one or a few better trails (solutions constructed)
- All pheromone values are usually subjected to "evaporation" by deducting a certain amount

Pheromone Update

- In general, two kinds of pheromone update strategies
  - Step-wise and trail-wise strategies
- Step-wise strategy
  - Each ant is granted the right to drop pheromone on each selection step
  - The Ant System, Ant Colony System adopt this strategy
- Trail-wise strategy
  - Allows one or a few ants to update the segments of their trails
  - The Ranked Ant System and Ant Colony System use this strategy

Pheromone Item Extraction

- Using a solution S to update the pheromone map, an operation is required to extract the mapped pheromone items from the solution
- Segmentation operation
  - Extract link between two objects or an object and a group

$$\boldsymbol{L} = \left\{ l_{ij} \middle| i, j = 1, 2, \ldots, z \right\} \text{ or } \boldsymbol{L} = \left\{ l_{ij} \middle| i = 1, 2, \ldots, z;\ j = 1, 2, \ldots, g \right\}$$

$$\mathrm{H}^{problem\ type}\left(\cdot\right)$$

$$\mathrm{H}^{problem\ type}\left(\mathbf{S}\right) \subseteq L$$

Primary Operations in ACO

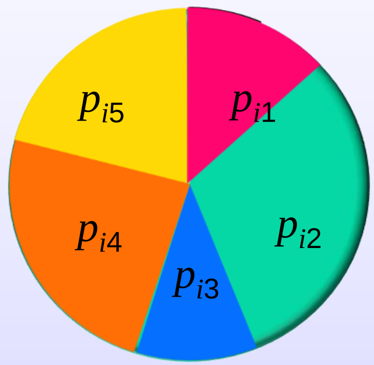
- Constructing a solution
  - (1) Update a set of candidate objects
  - (2) Selecting an object from the set
  - (3) Add the object to the partial solution
  - (4) Repeat steps1-3 until all objects are processed

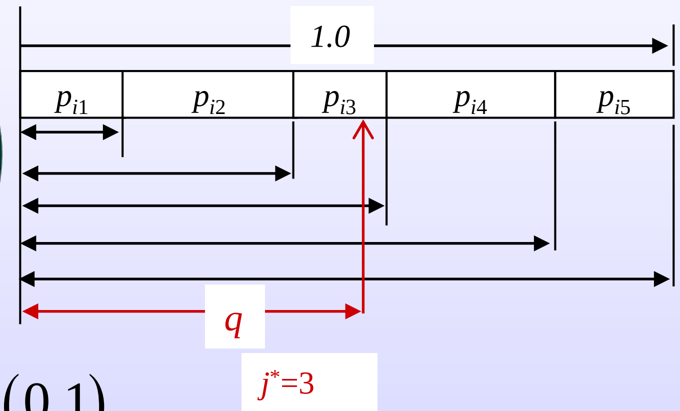
ACO algorithm

1. Initialize pheromone map.
2. Construct an initial solution and assign it the best solution so far.
3. For *r* ⟵ 1 to *iteration_limit*
4. For *k* ⟵ 1 to *number_of_ants*
5. For *i* ⟵ 1 to *number_of_objects*
6. Construct a candidate set of unprocessed objects.
7. Select one object from the candidate set and add it to the solution
8. If a step-wise pheromone update strategy is used, add pheromone to the related segment.
9. End for
10. If a trail-wise pheromone update strategy is used, add pheromone to the segments of the constructed trail.
11. End for
12. Execute the local search algorithm to improve solution qualities
13. Evaluate the constructed solution and update the best solution so far.
14. Check for stop conditions to exit the algorithm.
15. If elitism pheromone dropping strategy is used, identify the elite trail and add pheromone to it.
16. Subtract a certain amount of pheromone from each segment of pheromone to conduct pheromone evaporation.
17. If pheromone reset policy is used, check stagnation to reset pheromone map.
18. End for

Ant System

- AS, the first ACO system (Dorigo et al., 1991;1996）
- Selection probability for selecting object j to succeed object i





$$p_{ij} = \frac{\tau_{ij}^{\ \alpha} . \eta_{ij}^{\ \beta}}{\sum_{k \in N_i} \tau_{ik}^{\ \alpha} . \eta_{ik}^{\ \beta}}, \quad \forall j \in N_i$$

$$j^* = k, \text{where} \sum_{j=1}^{k} p_{ij} \ge q > \begin{cases} 0, k = 1 \\ \sum_{j=1}^{k-1} p_{ij}, 1 < k \le \left| N_i \right| \end{cases} \text{ and } q \sim U\left(0,1\right)$$

- j^* is the selected object
- N_i is a set of selectable candidate objects at the construction stage right after object i is processed or selected

- τ_{ij} is the pheromone value from object i to j
  - Pheromone values store the solution construction experience
- η_{ij} is computed from a heuristic evaluation for associating object i with j subject to the optimization goal
  - Must reflect the goal of optimization
  - Evaluation can be done by a deterministic heuristic method or greedy method
- AS uses roulette wheel method to stochastically select an object to proceed the solution construction

AS

- **Pheromone Update**
  - Pheromone map is updated after all ants have constructed their solutions
  - All ants are allowed to update the pheromone

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \Delta\tau_{ij}, \forall ij$$

$$\Delta\tau_{ij} = \sum_{k=1}^{y} \Delta\tau_{ij}^{k}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{f(\mathbf{S}^k)}, & \text{if } l_{ij} \in \mathrm{H}^{\text{problem type}}(\mathbf{S}^k) \\ 0, & \text{otherwise} \end{cases}$$

- ρ is the evaporation rate $0 \leqq \rho < 1$, Q is a constant, $f(S^k)$ is the objective function value of solution S^k constructed by ant k, $\mathrm{H}^{\textit{problem type}}()$ is the segmentation operator

Ant Colony System

- To improve Ant-Q system by Dorigo et al. (1997）
- Deterministic mode is added in computation of selection probability, in addition to the AS selection

$$j^* = \begin{cases} \arg \max\limits_{j \in N_i} \left\{ \tau_{ij}{}^{\alpha} . \eta_{ij}{}^{\beta} \right\}, \text{ if } \theta \le \theta_0 \\ k, \text{where} \sum\limits_{j=1}^{k} p_{ij} \ge q > \begin{cases} 0, k = 1 \\ \sum\limits_{j=1}^{k-1} p_{ij}, 1 < k \le |N_i| \end{cases}, \text{ otherwise.} \end{cases}$$

$$\theta, q \sim U(0,1)$$

$$p_{ij} = \frac{\tau_{ij}{}^{\alpha} . \eta_{ij}{}^{\beta}}{\sum\limits_{k \in N_i} \tau_{ik}{}^{\alpha} . \eta_{ik}{}^{\beta}}$$

- θ_0 is a threshold for either use deterministic or AS selection method (stochastic), $0 \leqq \theta_0 \leqq 1$

Ant Colony System

- **Step-wise pheromone update strategy is used for each ant**
  - **When an ant selects object j to succeed object $\imath$, the pheromone segment from i to j is updated**

$$\tau_{ij} \leftarrow (1-\sigma)\cdot\tau_{ij} + \sigma\cdot\Delta\tau_{ij}$$

$$\Delta\tau_{ij} = \begin{cases} \tau_0, \text{for ACS system} \\ \gamma\cdot\max_{k\in N_j}\{\tau_{jk}\}, \text{for Ant-Q System(Gambardella et al., 1995)}, 0 \le \gamma < 1 \end{cases}$$

  - **σ is the segment evaporation rate;** $0 < \sigma < 1$

- In each iteration only the best solution so far is allowed to update pheromone

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij}, \forall ij$$

$$\Delta\tau_{ij} = \begin{cases} \dfrac{1}{f(\mathbf{S}^*)}, & \text{if } l_{ij} \in \mathrm{H}^{\textit{problem type}}(\mathbf{S}^*) \\ 0, & \text{otherwise} \end{cases}$$

- ρ is the evaporation rate $0 \leqq \rho < 1$, $f(\mathrm{S}^*)$ is the objective function value of the best solution so far S^*, $\mathrm{H}^{\textit{problem type}}()$ is the segmentation operator

AS-Rank System

- Proposed by Bullnheimer et al.（1997）
- A few elite ants are allowed to update pheromone and the dropping amounts are proportional to the ranks
  - The weight of pheromone amount added by k ranked ant is $w+1-k$
- Selection mechanism is the same as AS

$$f^1, f^2, \ldots, f^p \text{ are objective function values}$$

$$f'^1 \le f'^2 \le \ldots \le f'^p, \text{ are sorted values}$$

$$f^{a_k} = f'^k, a_k \in \{1, 2, \cdots, p\}$$

f^q is the objective function value of the solution constructed by ant q; f'^k is the k ranked objective function value; p is the total number of ants

- ω is the number of elite ants; a_k is k ranked ant; S^q is the solution constructed by ant q ; S^* is the best solution so far; Q is a constant

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \Delta\tau_{ij} + \Delta\tau_{ij}^{\;*}, \forall ij$$

$$\Delta\tau_{ij} = \begin{cases} \sum_{k=1}^{\omega}(\omega+1-k)\dfrac{Q}{f^{a_k}}, \text{if } l_{ij} \in \bigcup_{k=1}^{\omega}\left\{\mathrm{H}^{\textit{problem type}}\left(\mathbf{S}^{a_k}\right)\right\} \\ 0 \qquad\qquad , \text{otherwise} \end{cases}$$

$$\Delta\tau_{ij}^{*} = \begin{cases} (\omega+1)\dfrac{Q}{f(\mathbf{S}^{*})}, \quad \text{if } l_{ij} \in \mathrm{H}^{\textit{problem type}}(\mathbf{S}^{*}) \\ 0 \qquad\qquad , \text{otherwise} \end{cases}$$

MMAS, MAX-MIN Ant System

- Proposed by Stützle and Hoos (1997) to avoid pheromone over dropping
- Limit the pheromone value within a lower and a upper bound

$$\tau_{\min} \leq \tau_{ij} \leq \tau_{\max}$$

- Pheromone update is executed once in each iteration
  - Alternatively use iteration best solution and the best solution so far to drop pheromone
  - Purpose is to design and control the pheromone maturity

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \Delta\tau_{ij}, \forall ij$$

$$\Delta\tau_{ij} = \begin{cases} \dfrac{1}{f\left(\mathbf{S}^*\right)}, & \text{if global best update is used and } l_{ij} \in \mathrm{H}^{\textit{problem type}}(\mathbf{S}^*) \\ \dfrac{1}{f\left(\mathbf{S}'\right)}, & \text{if iteration best update is used and } l_{ij} \in \mathrm{H}^{\textit{problem type}}(\mathbf{S}') \\ 0, & \text{otherwise} \end{cases}$$

- S^* **is the best solution so far;** S' **is the iteration best solution**
- **For the first 100 iterations, the iteration best solution is used to update pheromone except for every 5 iterations the best solution so far is used instead**

- After 100 iterations, the frequency of using the best solution so far is increased to every 3 iterations
- Pheromone value is subject to the upper and lower bounds; i.e.,

$$\tau_{\min} \le \tau_{ij} \le \tau_{\max}, \tau_{\min} > 0$$

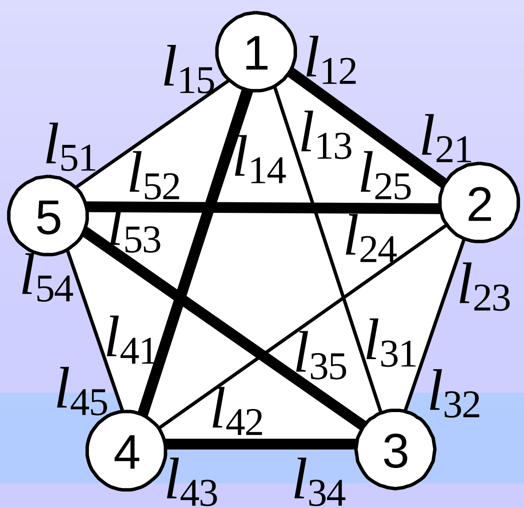
$$\text{if } \tau_{ij} > \tau_{\max}, \tau_{ij} \leftarrow \tau_{\max}.$$

$$\text{if } \tau_{ij} < \tau_{\min}, \tau_{ij} \leftarrow \tau_{\min}.$$

Pheromone Maturity and Search Stagnation

- When the pheromone map is over maturated, the solution search is stagnated
- A branch factor is proposed to evaluate the maturity of the pheromone map
- When a pheromone is over mature
  - A local optimum is significantly stored in the map
  - A biased trail can be detected from the map

$$\mathbf{T}=\begin{bmatrix} 0.0 & \mathbf{0.9} & 0.2 & \mathbf{0.8} & 0.1 \\ \mathbf{0.9} & 0.0 & 0.2 & 0.1 & \mathbf{0.7} \\ 0.2 & 0.2 & 0.0 & \mathbf{0.8} & \mathbf{1.0} \\ \mathbf{0.8} & 0.1 & \mathbf{0.8} & 0.0 & 0.1 \\ 0.1 & \mathbf{0.7} & \mathbf{1.0} & 0.1 & 0.0 \end{bmatrix}$$



$$\mathbf{T} \equiv [\tau_{ij}]_{z\times z} = \begin{bmatrix} \tau_{11} & \tau_{12} & \cdots & \tau_{1z} \\ \tau_{21} & \tau_{22} & \cdots & \tau_{2z} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{z1} & \tau_{z2} & \cdots & \tau_{zz} \end{bmatrix}$$

$$\tau_i^{\min} = \min_{j=1,\ldots,z} \tau_{ij}, \tau_i^{\max} = \max_{j=1,\ldots,z} \tau_{ij}$$

$$\chi_i = \tau_i^{\min} + \kappa(\tau_i^{\max} - \tau_i^{\min}), 0 \le \kappa \le 1$$

$$B = \frac{\sum_{i=1}^{z}\sum_{j=1}^{z}\omega_{ij}}{2Z}, \omega_{ij} = \begin{cases} 1, if\ \tau_{ij} \ge \chi_i \\ 0, otherwise \end{cases}.$$

- κ is an interpolation factor
- When two values are relatively larger than others in each row, they will be filtered out by the thresholds, and the branch factor of the map will approach 1.0
- A pheromone map becomes mature when its branch factor is close to 1.0

ACO for Traveling Salesman Problems

- TSP → object-sequencing problem
  - Vehicle routing, integrated-circuit chip placement, assembly sequencing, job scheduling, etc
- The objective of the TSP is to minimize the route length

Selection Probability of TSP

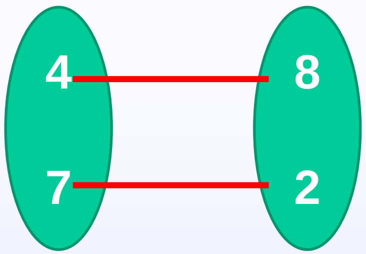
- Power factors α and β are used to intensify or relieve the influence of the values of τ_{ij} and η_{ij}, respectively
- A higher value of α → stronger probability that object j will be chosen to succeed object i
- If τ_{ij} dominates the value, the solution search relies heavily on previous construction experience
  - Search stagnation easily occurs
    - All ants refer to the same pheromone map and the map is likely to be biased with a dominating route

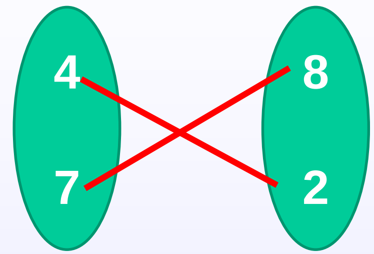
- if η_{ij} the heuristic value dominates the selection
  - The solution search turns out to be a "greedy search"

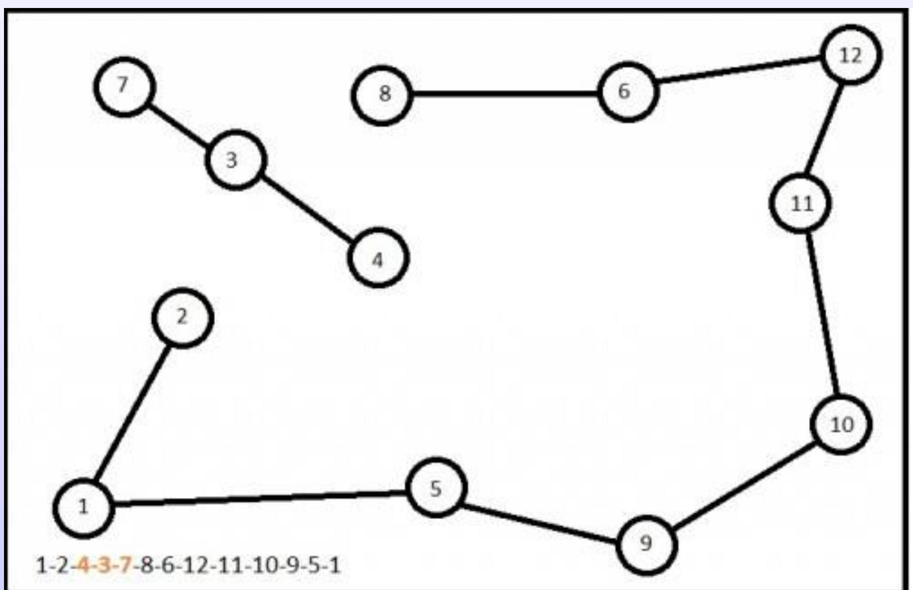
Local Search for Object Sequencing Problems

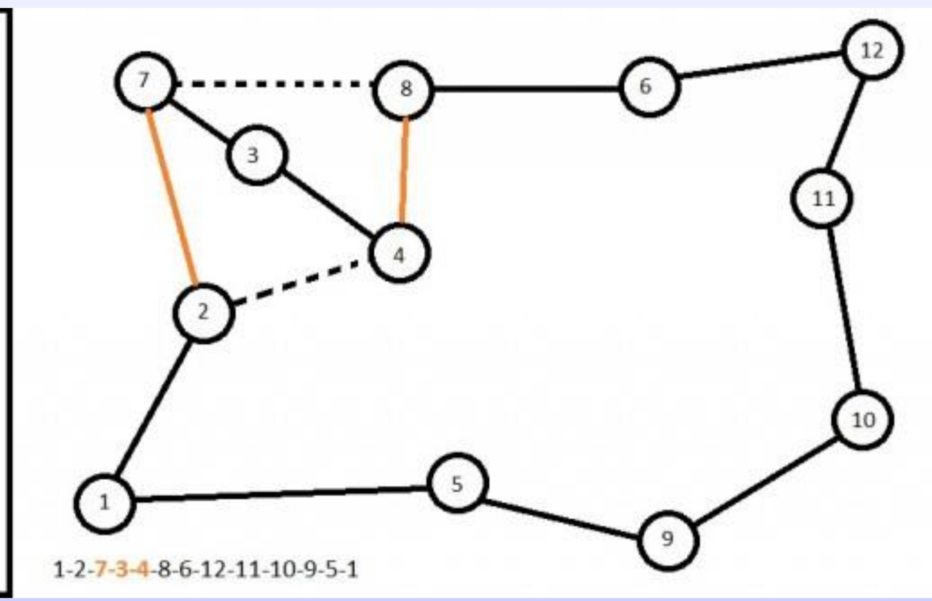
- The *K-Opt* local search method was originally presented by Lin, S. (1965) to improve the computed routes of TSPs
- Much research in the literature uses *2-Opt* and *3-Opt* methods to improve the constructed object sequences
- These local search methods are usually conducted stochastically for a finite number of trials

2-opt

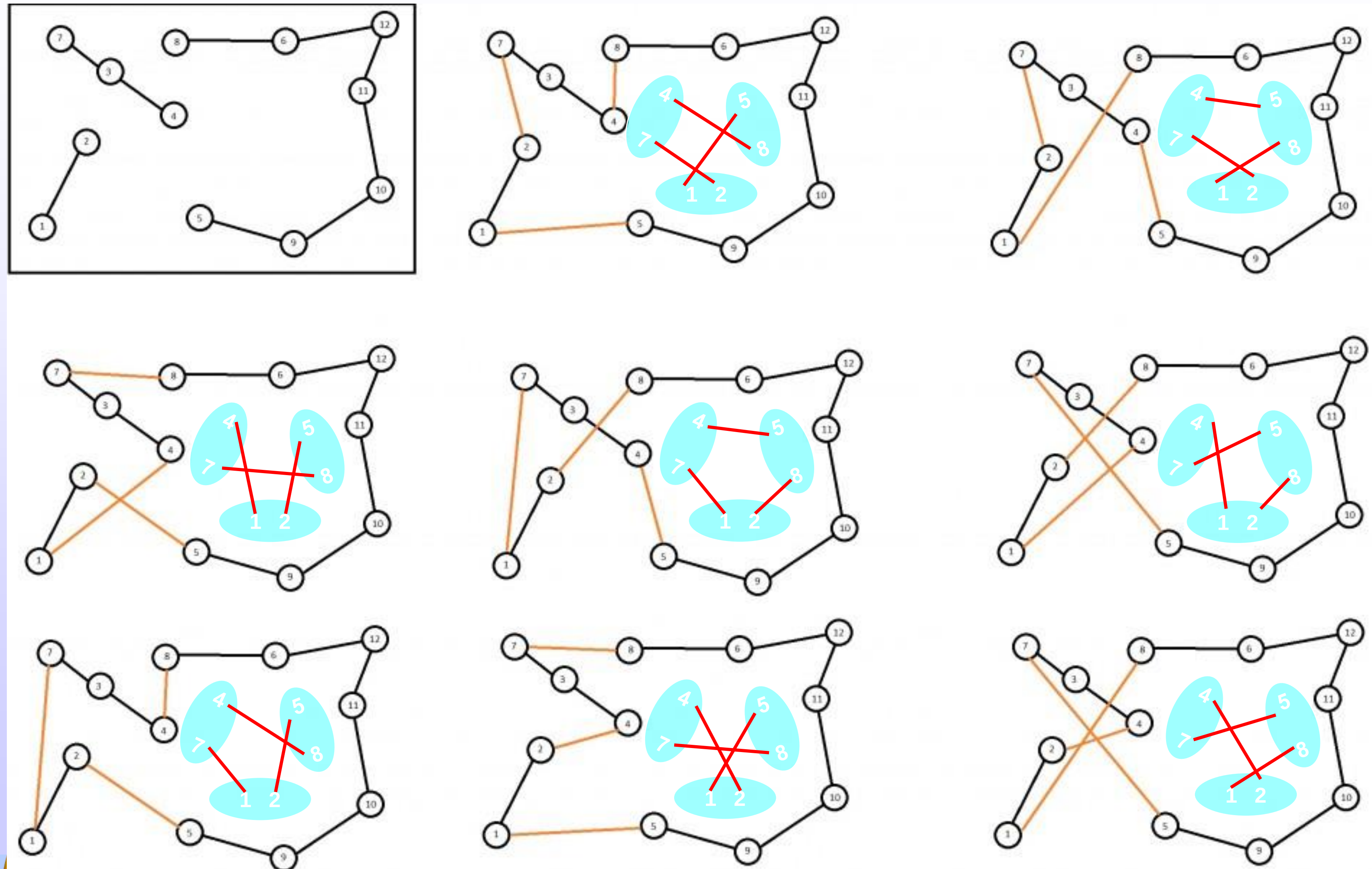








3-opt



- Once an improved solution is found or the limit of the number of trials is reached, the local search operation is terminated
- Ant Colony System (ACS) used a *K-Opt* local search to solve large-scale TSPs
  - The proposed system thus generates high-quality solutions for large-scale TSPs
  - However, solution-search stagnation is generally not avoided

ACO for Bin-packing Problems (BPPs)

- Regarded as object-grouping problems
- One-dimensional bin-packing problem is to pack weighted objects (boxes or parcels) into a number of bins, subject to a weight capacity constraint of the bin
  - The goal is either to group these objects into a minimum number of bins or to minimize the weight variances between the bins
- Levine and Ducatelle (2003) proposed an object-object pheromone map for BPPs

$$p_{aj} = \begin{cases} \dfrac{\tau_a(j)^{\alpha}\,\eta(j)^{\beta}}{\sum_{g \in N_a} \tau_a(g)^{\alpha}\,\eta(g)^{\beta}}, & \text{if } j \in N_a \\ 0, & \text{otherwise} \end{cases}, \quad \tau_a(j) = \begin{cases} \dfrac{\sum_{i \in S_a} \tau_{ij}}{|S_a|}, & \text{if } S_a \neq \varnothing, \\ 1 & \text{,otherwise} \end{cases}$$

$$S_a = \left\{ s_{i1}, s_{i2}, \ldots, s_{i|S_a|} \right\},$$

- N_a **is the set of unprocessed objects whose weight does not exceed the weight capacity left in bin** a
- $\tau_a(j)$ **is the average pheromone values of object** j **and packed objects in bin** a
- S_a **is the set of objects currently packed in bin** a
- P_{aj} **is the probability of choosing object** j **to pack into bin** a
- $\eta(j)$ **is simply the weight of object** j

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \Delta\tau_{ij}, \forall ij, \Delta\tau_{ij} = \begin{cases} \dfrac{\rho}{f(\mathbf{S}^*)} & \text{, if } l_{ij} \in \mathrm{H}^{BPP}(\mathbf{S}^*) \\ 0 & \text{, otherwise} \end{cases}$$

$$\mathrm{H}^{BPP}(\mathbf{S}) = \bigcup_{a=1}^{g} \left\{ l_{pq} \mid p, q \in S_a \right\}$$

- **Only the best solution so far is granted rights for pheromone update**
- **Object-object links are extracted from each bin to update the pheromone items**

ACO Applications and Research

Problems	Year	Authors	Algorithm Name
Vehicle routing	1996 1999	Bullnheimer and Strauss Gambardella et. al.	AS-VRP MACS-VRPTW
Network routing	1998 1998 2001	Di Caro and Dorigo Bonabeau et. al. Baboglu O. et. al.	AntNet-FS ABS-smart ant Anthill
Sequential ordering	1997	Gambardella and Dorigo	HAS-SOP
Graph coloring	1994 1997	Costa and Hertz Costa and Hertz	ANTCOL ANTCOL
Frequencing assignment	1998	Maniezzo and Carbonaro	ANTS-FAP
Generalized assignment	1998	Ramalhinho and Serra	MMAS-GAP
Bin Packing & Cutting Stock	2003	Levine, J. and F. Ducatelle	MMAS-BIN

ACO Applications and Research

Problems	Year	Authors	Algorithm Name
Traveling Salesman	1991 1995 1997 1997 1997 2000	Dorigo et. al. Gambardella and Dorigo Dorigo and Gambardella Stützle and Hoos Bullnheimer et. al. Cordon, et. al.	BWAS
Quadratic Assignment	1994 1997 1999	Maniezzo et. al. Taillard and Gambardella Stützle and Hoos	AS-QAP FANT MMAS-QAP
Scheduling	1994 1997 1999 1999 2001 2002	Colorni et. al. Bus driver scheduling Bauer et. al. Zwaan and Marques Cicirello, V.A Blum and Sample	AS-JSP AS ACS-SMTTP ACS-JSP AC^2 ACO-FOP

Discussion

- How to solve continuous optimization problems?
  - Hierarchically discretize the solution space into finite number of subdomains?

ACO Demo

DEMO

References

- Bullnheimer, B., R. F. Hartl, et al. (1997). A New Rank Based Version of the Ant System A Computational Study Vienna, University of Vienna, Austria.
- Dorigo, M. and L. M. Gambardella (1997). "Ant colony system: a cooperative learning approach to the traveling salesman problem." Evolutionary Computation, IEEE Transactions on 1(1): 53-66.
- Dorigo, M., V. Maniezzo, et al. (1996). "Ant system: optimization by a colony of cooperating agents." Systems, Man and Cybernetics, Part B, IEEE Transactions on 26(1): 29-41.
- Dorigo, M. and T. Stützle (2001). The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances Handbook of Metaheuristics.
- Falkenauer, E. (1996). "A hybrid grouping genetic algorithm for bin packing." Journal of Heuristics 2(1): 5-30.
- Gambardella, L. M. and M. Dorigo (1995). Ant-Q: A Reinforcement Learning approach to the traveling salesman problem Proceedings of ML-95, Twelfth International Conference on Machine Learning
- Gambardella, L. M. and M. Dorigo (1996). Solving symmetric and asymmetric TSPs by ant colonies.
- Levine, J. and F. Ducatelle (2003). Ants Can Solve Difficult Bin Packing Problems. Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2003), Nottingham, UK, August 13-16th.
- Lin, S. (1965). Computer solution of the traveling salesman problem.
- Stutzle, T. and H. Hoos (1997). MAX-MIN Ant System and local search for the traveling salesman problem.
- Stutzle, T. and H. Hoos (2000). "MAX-MIN Ant system." Future Generation Computer Systems 16(9): 889 - 914.

Programming Assignments

- Design an ACO system to solve either JSPs or assembly sequence planning problems or TSPs
  - AS
  - ACS
  - AS_Rank
  - AS_Elite
- You can read related paper and redo their work