Applications of Ant Colony Optimization Techniques in Project Scheduling

Presented by:

Avirup Guha Neogi

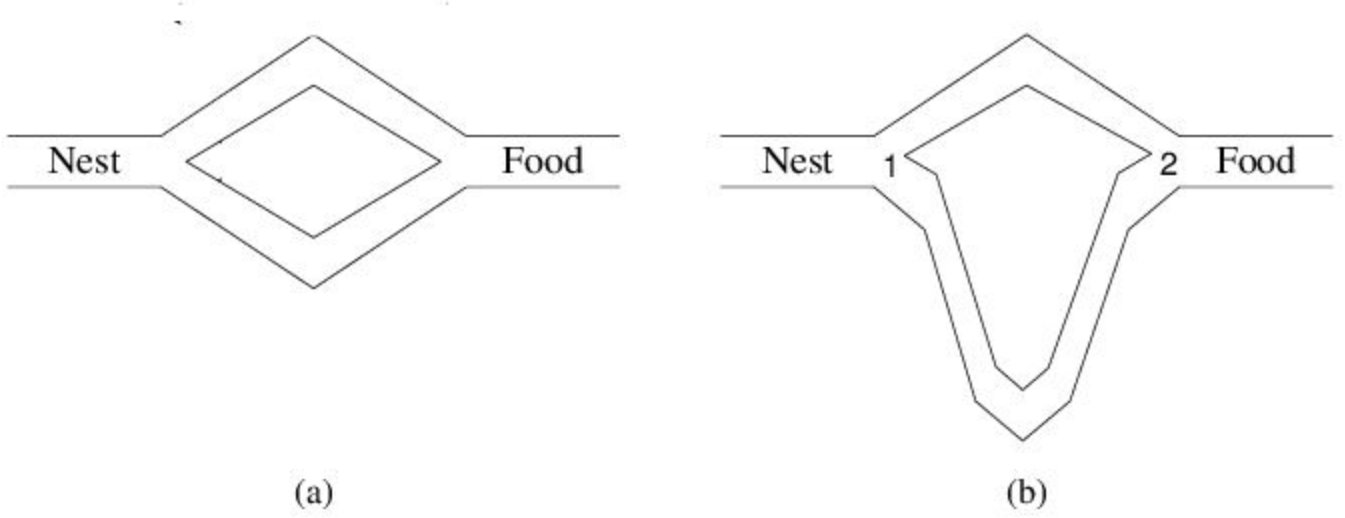
Bonn-Aachen International Center for Information Technology, Bonn

Introduction

- Probabilistic technique.
- Searching for optimal path in the graph based on behaviour of ants seeking a path between their colony and source of food.
- Meta-heuristic optimization

ACO Concepts

- Ants navigate from nest to food source. Ants are blind!
- Shortest path is discovered via pheromone trails.
- Each ant moves at random Pheromone is deposited on path.
- More pheromone on path increases probability of path being followed



Mathematical Model of Combinatorial Optimization Problem (COP)

- Let us consider minimization problem : (S,f,Ω)
- The goal is to find a *globally optimal* solution $s_{opt} \in S$
- A finite set $C = \{c_1, c_2,...,c_N\}$ of components is given.
- The states of the problem are defined in terms of sequences $x = c_i, c_j, ..., c_k, ...$ over the elements of C. The set of all possible sequences is denoted by X . The length of a sequence x, i.e., the number of components in the sequence, is expressed by $|x|$.
- The finite set of constraints Ω defines the set of feasible states $\tilde{X}$, with $\tilde{X} \subseteq X$.
- A set s* of feasible solutions is given, with $s^* \subseteq \tilde{X}$ and $s^* \subseteq S$.
- A cost f(s, t) is associated to each candidate solution $s \in S$.

Mapping ACO to COP

- Ants build solutions on the construction graph G=(C,L)
- Ants solution construction policy includes associated pheromone information τ_{ij} and heuristic information η_{ij}
- It exploits the graph G = (C,L) to search for feasible solutions s of minimum cost. That is, solutions s such that $f^*_s = \min_s f(s, t)$
- It has a memory M^k that it uses to store information about the path it followed so far. Memory can be used (i) to build feasible solutions (i.e., to implement constraints Ω), (ii) to evaluate the solution found, and (iii) to retrace the path backward to deposit pheromone.
- It can be assigned a start state x^k_s and one or more termination conditions e^k. Usually, the start state is expressed either as a unit length sequence (that is, a single component sequence), or an empty sequence

Mapping ACO to COP contd…

- When in a particular state, , if no termination condition is satisfied, it moves to a node j in its neighbourhood N^k_i . Often, moves towards feasible states are favoured, either via appropriately defined heuristic values η, or through the use of the ants' memory.
- It selects the move by applying a probabilistic decision rule. Its probabilistic decision rule is a function of (i) locally available pheromone trails and heuristic values, (ii) the ant's private memory storing its past history, and (iii) the problem constraints
- The construction procedure of ant k stops when at least one of the termination conditions e^k is satisfied.
- When adding a component c_j to the current solution it can update the pheromone trail associated to it or to the corresponding connection. This is called online step-by-step pheromone update.
- Once built a solution, it can retrace the same path backward and update the pheromone trails of the used components or connections. This is called online delayed pheromone update

Algorithmic Formulation of ACO

Procedure *ACO Metaheuristic*

Schedule Activities

ManageAntsActivity()

UpdatePheromones()

DaemonActions() {Optional}

End Schedule Activities

End *ACO Metaheuristic*

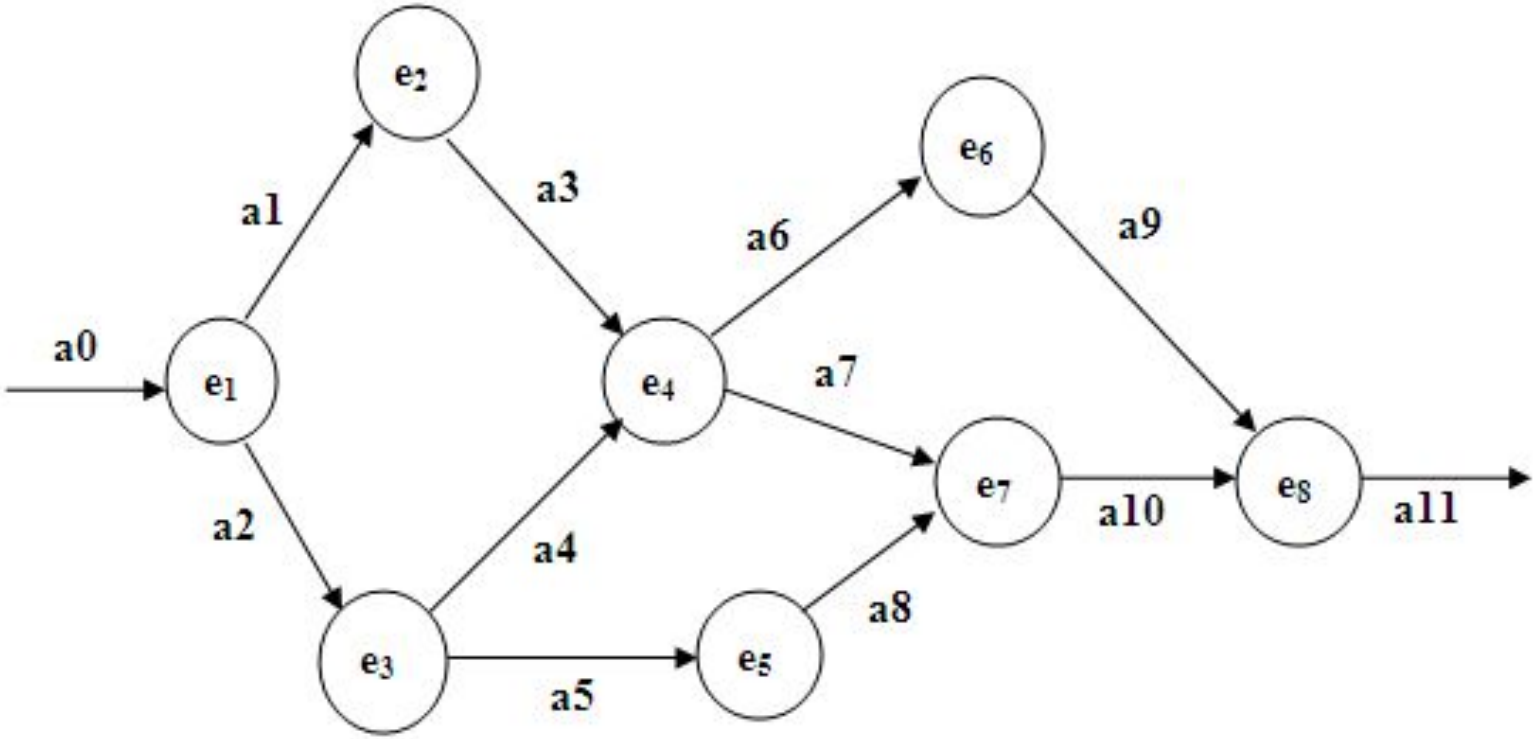
Historical developments of ACO variants

- Ant System was developed by Marco Dorigo in his PhD thesis in 1992.
- Elitist Ant System was developed by Marco Dorigo in 1992
- Max-Min Ant System developed by Hoos and Stützle in 1996
- Ant Colony System was developed by Gambardella Dorigo in 1997
- Rank-based Ant System was developed by Bullnheimer, Hartl, & Strauss in 1997

Project Scheduling Problem

- **Scheduling** can be critical to the successful completion of many engineering projects.
- Some activities can proceed simultaneously, and some must occur before or after other activities.
- Completing a project in the shortest possible time depends on properly scheduling these activities, identifying the critical activities that must be completed on schedule, avoiding unacceptable fluctuations in resource requirements.
- **Project Scheduling** involves arranging a set of Activities into a diagram, table, or chart that shows when each Activity will start and stop.
- A major concern is that the set of Activities is subject to precedence constraints, specifying which tasks have to be completed before other tasks can start.

Activitv diagram



The Figure represents the precedence relationship among different activities.

Problem Formulation

Construction Graph

The underlying problem is represented in terms of a graph, G = <N,E>.

The graph is connected, but not necessarily complete, such that the feasible solutions to the original problem correspond to paths on the graph which satisfy precedence constraints.

Then each ant will perform a tour on the graph by constructing a path according to the node transition rule described next.

Constraints

The only constraint in project scheduling is precedence constraint. The ants move from node to node satisfying the precedence constraint.

Problem Formulation contd…

Heuristic Information

The heuristic information η_{ij} is typically inversely proportional to the time taken to perform an activity between two events

Node transition rule

The ants walk through the construction graph based on a node transition rule. According to the precedence constraints, some nodes could be marked as inaccessible for a walking ant. The node transition rule is probabilistic. For the kth ant on node i, the selection of the next node j to follow is according to the node transition probability:

$$p_{ij}^{k} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{h \notin tabu_k} (\tau_{ih})^{\alpha}(\eta_{ih})^{\beta}} & if \quad j \notin tabu_k \\ 0 & otherwise \end{cases}$$

Problem Formulation contd…

Pheromone updating rule

The ant move from node to node by iteratively applying the node transition rule until a solution to the original problem is constructed. We define that a cycle of the AC algorithm is completed when every ant has constructed a solution. At the end of each cycle, the intensity of pheromone trails on each edge is updated by the pheromone updating rule:

$$\tau_{ij} \leftarrow \rho\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

ACO Algorithm for project scheduling

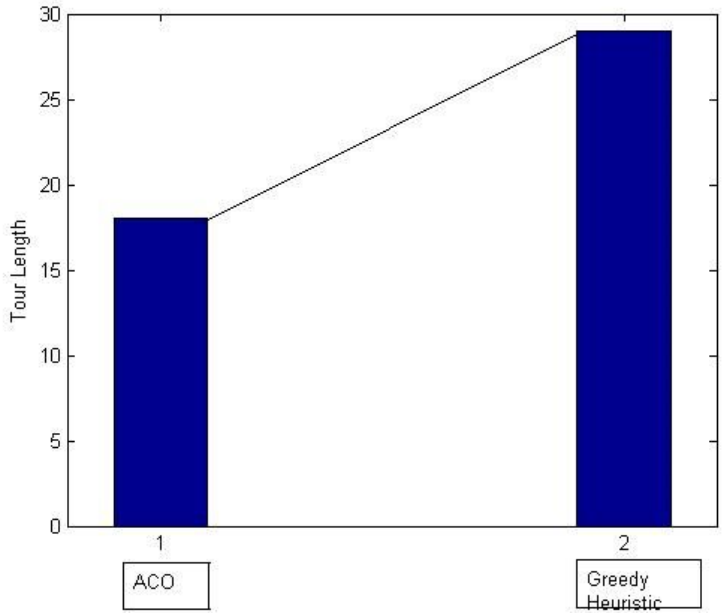
- Step 1: We construct the incidence matrix from the graph.
- Step 2: We calculate event predecessors of events.
- Step 3: We calculate event successors of events.
- Step 4: From steps 2 and 3, compute the precedence relations of activities.
- Step 5: We compute the neighbourhoods of all the events considering the precedence relations.
- Step 6: We initialize pheromone value to 1.
- Step 7: We assign Heuristic information to all the nodes.

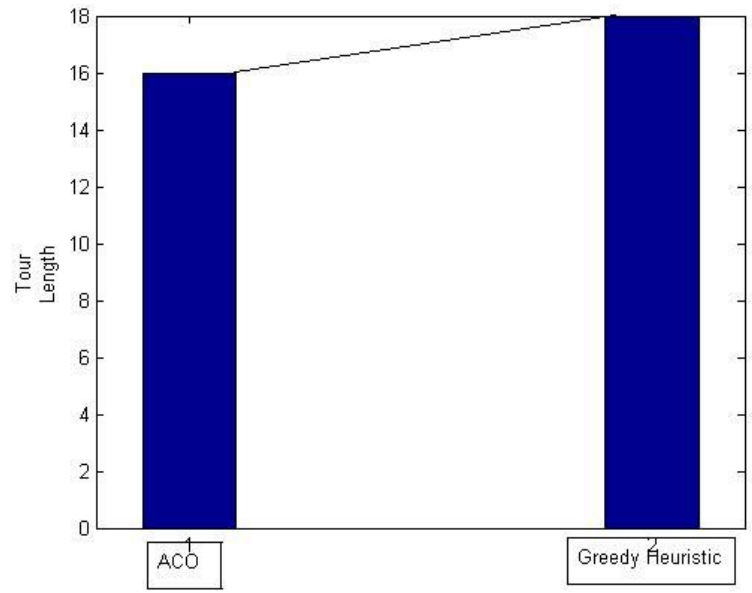
ACO Algorithm for project scheduling contd…

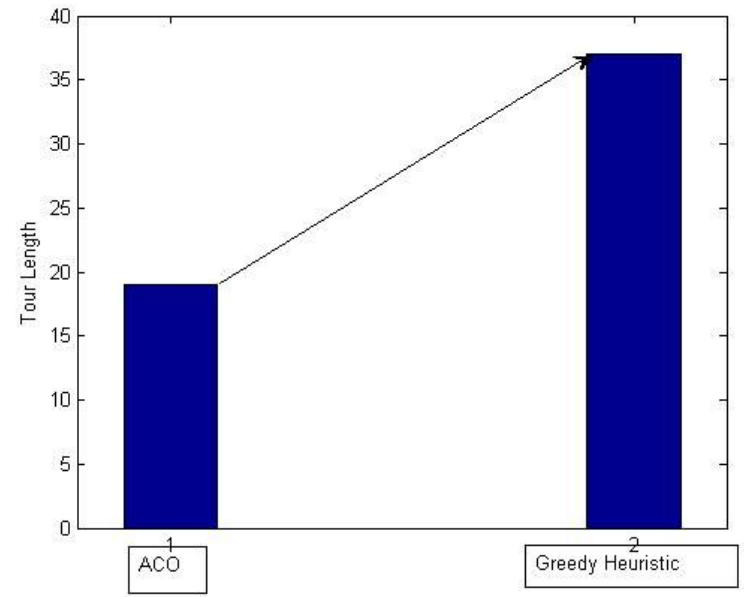
- Step 8: We construct the probability matrix by using the Roulette Wheel selection method.
- Step 9: Ant is initialized
- Step 10: We select the event j from event i having the maximum probability.
- Step 11: After a complete tour, pheromone is evaporated from the formula given above.
- Step 12: We then add pheromones on the arcs traversed by the ant by the formula discussed earlier
- Step 13: Length of the tour is computed.
- Step 14: The procedure is run until the stopping criteria is met.

.

Results


Tour Length
30
25
20
15
10
5
0
1
2
ACO
Greedy Heuristic


Tour Length
18
16
14
12
10
8
6
4
2
0
1
2
ACO
Greedy Heuristic


Tour Length
40
35
30
25
20
15
10
5
0
1
2
ACO
Greedy Heuristic

Conclusion and Future Scope

- ACO outperforms greedy heuristic algorithms
- ACO can be coupled with other local search and soft computing approaches to further optimize the solution

References

[1] M. Dorigo, V. Maniezzo and A. Colorni. (1991). Positive feedback as a search strategy, Tech. Rep.91-016. Politecnico di Milano.

[2] M. Dorigo. (1992). Optimization, learning and natural algorithms. Ph.D. Thesis, Dip. Elettronica e Informazione, Politecnico di Milano, Italy.

[3] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. (1990). The self-organizing exploratory pattern of the Argentine ant. Journal of Insect Behavior(vol. 3, p. 159).

[4] W. Herroelen, B. D. Reyck, and E. Demeulemeester, Resource-constrained project scheduling, a survey of recent developments, *Computer Oper. Res.*, vol. 13, no. 4, pp. 279–302, 1998.

[5] P. Brucker, A. Drexl, R.Mohring, K. Neumann, and E. Pesch, Resource constrained project scheduling: Notation, classification, models and methods, *Eur. J. Oper. Res.*, vol. 112, pp. 3–41, 1999.

Thank You !!