OAKLAND UNIVERSITY

Project 1: Ant Colony Optimization Algorithms

Adam DeJans

March 1, 2017

1 INTRODUCTION

Combinatorial optimization problems naturally arise in the industrial world all the time. Often, having sufficient solutions to these problems can save companies millions of dollars. We know of many examples in which optimization problems arise; for example: bus scheduling, telecommunication network design, travelling, and vehicle routing problems. Perhaps the most famous combinatorial optimization problem is the travelling salesman problem (TSP). The TSP can be simply thought of as the problem of figuring out a tour of cities a salesman must travel (visiting each city exactly once) so that the total distance travelled is minimized.

With the evergrowing arise of combinatorial optimization problems and their intrinsic link to industrial problem, many researchers, mathematicians and computer scientists, have developed a plethora of algorithms to solve these problems. We now distinguish the two types of algorithms; *complete* and *approximate*. Complete algorithms are able to solve these problems in such a way that in the end, the optimal solution is given. However, since many of these problems are NP-hard the optimal solution may take a long time to obtain. This is the reason why there are approximate algorithms. Approximate algorithms are designed, as expected, to give approximately the optimal solution. The great thing about using an approximate algorithm is that they can obtain good results in a relatively short amount of time.

Ant colony optimization (ACO) algorithms are some of the most recent class of algorithms designed to approximate combinatorial optimization problems. The algorithm behaves similar to real ants and their biological abilities to find the nearest food source and bring it back to their nest. The main source of communication between ants is the depositing of chemically produced pheromone onto their paths. It is with this key idea that Marco Dorigo and colleagues were inspired to create this new class of algorithms.

2 SIMULATED VS. REAL ANTS

While Ant Algorithms are based off of real colonies of ants cooperating as individuals using pheromone trails and stigmergy to find shortest paths, there are obviously several differences between real and simulated ants in ACO algorithms. Artificial ants traverse a discrete graph and thus take discrete steps when visiting vertices. So unlike real ants, the artificial ants can only deposit pheromone at vertices in the graph after the ant has moved[1].

3 MODELING THE TSP WITH ACO ALGORITHMS

3.1 ANT SYSTEM FOR THE TSP: THE FIRST ACO ALGORITHM (1991)

[Dorigo, 1996]

3.1.1 TRANSITION PROBABILITY

The transition probability from city i to city j for the k-th ant is defined as

$$p_{i,j}^k(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,j}(t)]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{i,l}(t)]^\alpha \cdot [\eta_{i,l}(t)]^\beta} & if\ j \in \mathcal{N}_i^k \\ 0 & otherwise \end{cases}$$

and we note that the fraction ensures that $\sum_{j \in V} p_{i,j}^k(t) = 1$.

Variables in the formula are defined as follows:

- $\tau_{i,j}(t)$ = amount of pheromone trail on edge (i, j) at time t.
- $\eta_{i,j}$ = the visibility; expressed as $\frac{1}{d_{i,j}}$, where $d_{i,j}$ is the euclidean distance between two vertices.
- α = parameter controlling the relative weight of phermone trail
- β = parameter controlling the weight of visibility

Data-strutures that are used are:

- $tabu_k$ – dynamically growing vector containing the tabu list of the k-th ant (list of vertices visited by the k-th ant)
- $\mathcal{N}_i^k = V \setminus tabu_k$ – list of vertices not visited by the k-th ant

[1] In many cases, ants are designed to update pheromone trails only after having generated a solution.

3.1.2 PHEROMONE UPDATE

After each ant has completed their tour, pheromone evaporation on all arcs is triggered, and then each ant k deposits a quantity of pheromone $\triangle\tau^k_{i,j}(t)$ on each arc that it has used.

$$\triangle\tau^k_{i,j}(t) = \begin{cases} \frac{Q}{L^k(t)} & if\ arc\ (i,j) \in tabu_k(t) \\ 0 & otherwise \end{cases}$$

Where Q is a constant[2] and $L^k(t)$ is the length of the tour done by ant k at iteration t.

One can implement the addition of new pheromone and pheromone evaporation to all arcs by

$$\tau_{i,j}(t) \leftarrow (1-\rho)\tau_{i,j}(t) + \triangle\tau_{i,j}(t)$$

where $\triangle\tau_{i,j}(t) = \sum_{k=1}^{m} \triangle\tau^k_{i,j}(t)$ and ρ is a constant in $(0,1]$ representing *evaporation.*

3.2 ANT COLONY SYSTEM (ACS) (1996)

Improved from the Ant System in 1996 [Dorigo, 1999]

3.2.1 TOUR CONSTRUCTION

$$j = \begin{cases} argmax_{l \in \mathcal{N}^k_i}\{\tau_{i,l}\left[\eta\right]^\beta\} & if\ q \le q_o\ \text{(Exploitation)} \\ J & otherwise\ \text{(Exploration)} \end{cases}$$

where q is a random variable uniformly distributed in $[0,1]$, q_o $(0 \le q_o \le 1)$ is a parameter, and J is a random variable selected according to the probability distribution

$$p^k_{i,j}(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,j}(t)]^\beta}{\sum_{l \in \mathcal{N}^k_i}[\tau_{i,l}(t)]^\alpha \cdot [\eta_{i,l}(t)]^\beta} & if\ j \in \mathcal{N}^k_i \\ 0 & otherwise \end{cases}$$

3.2.2 LOCAL TRAIL PHEROMONE UPDATE

Similar to evaporation we update pheromone on the trail by

$$\tau_{i,j}(t) \leftarrow (1-\rho)\tau_{i,j}(t) + \triangle\tau_{i,j}(t)$$

but we let $\triangle\tau_{i,j} = (nL_{nn})^{-1}$; where L_{nn} is the length of a nearest-neighbor tour.

[By doing the local pheromone update (results were improved and made great use of positive-feedback. –improved results, etc.) *******]

[2] was found to have negligible influence

3.2.3 GLOBAL PHEROMONE UPDATE

After each iteration the best ant is chosen to update it's tour by setting additional pheromone inversely proportional to the length of it's tour

$$\tau_{i,j}(t) \leftarrow (1-\rho)\tau_{i,j}(t) + \rho\frac{1}{L^{ba}} \quad \forall (i,j) \in T^{ba}$$

where L^{ba} is the length of the best ant's tour and T^{ba} is the tour of the best ant.

3.3 ANT-Q (1995)

Before ACS, there was Ant-Q. Essentially Ant-Q was an algorithm that attempted to merge main properties of both AS and Q-learning [Dorigo, 1999]. The only difference from ACS is that Ant-Q does the pheromone update following the rule

$$\tau_{i,j}(t) \leftarrow (1-\rho)\tau_{i,j}(t) + \rho\left[\gamma \cdot \max_{l \in \mathcal{N}_j^k} \tau_{j,l}\right]$$

It was eventually recognized that if the complicate prediction term was set to a small constant value (like that in ACS), then the algorithm had approximately the same performance. It was for this reason that ACS was chosen in place of Ant-Q; as ACS had approximately the same performance but was much simpler to implement [Dorigo, 1999].

4 ANT ALGORITHMS VS. OTHER TECHNIQUES FOR TSP

It can be shown that by way of dynamic programming the TSP can be solved in time $O(n^2 2^n)$. Much research has been put into improving the time but it has been proven to be difficult. As of now, it has not been determined whether an exact algorithm for TSP with runtime $O(c^n)$ (with $c < 2$) exists or not [Juenger, 2003].

While approximation algorithms may not give the optimal solution, they yield good solutions quickly. The computational complexity of the *ant-cycle* algorithm (a slight variation of AS) is $O(NC \cdot n^2 \cdot m)$ if the stopping criteria of the algorithm is to terminate after NC cycles; where n is the number of nodes and m is the number of ants [Dorigo, 1996].

4.1 COMPARISON TO TSP-TAILORED HEURISTICS

Ant-cycle showed great ability in finding good solutions; it either outperformed others or did just as good. However, *ant-cycle* compared to special-purpose heuristic's, the ant-algorithm took a significant amount more of computational time [Dorigo, 1996].

5 COMPARISON TO OTHER HEURISTIC APPROACHES

5.1 QUADRATIC ASSIGNMENT PROBLEM

The Quadratic Assignment Problem (QAP) (introduced in 1957 by Koopmans and Beckman) is an optimization problem in which the goal is to assign a set of n facilities to a set of n locations while minimizing the total assignment cost; where the "assignment cost" for any pair of facilities is a function of the flow between the facility and the distance between their locations (typically the sum of the distances multiplied by the corresponding flow values). QAP often arises when trying to model the placement of electronic components on a microchip. When closely examined we may think of the TSP as the case of QAP under the assumption that the flows connect all facilities along a single ring; i.e., the flows all have the same non-zero constant value.

A comparison of AS, and modified version of AS with an added non-deterministic hill climbing procedure, along with many other popular heuristics were compared in [Dorigo, 1996] against test problems known as Nugent problems[Nugent, 1968], Elshafe[Elshafei, 1977]i, and Krarup[Krarup, 1978]. In the table below from [Dorigo, 1996], it can be seen that AS always performed very well and AS was always within 5% of the best known. AS with local optimization (non-deterministic hill climbing) arrived at the best-known solution except for one problem (see table below).

Comparison of AS with other heuristic approaches. (results averaged over five runs. Best known results are in bold.)[Dorigo, 1996]

Algorithm/Problem	Nugent (15)	Nugent (20)	Nugent (30)	Elshafei (19)	Krarup (30)
Best known	**1150**	**2570**	**6124**	**17212548**	**88900**
AS	**1150**	2598	6232	18122850	92490
AS (non-deterministic hill climbing)	**1150**	**2570**	6128	**17212548**	**88900**
Simulated Annealing	**1150**	**2570**	6128	17937024	89800
Tabu Search	**1150**	**2570**	**6124**	**17212548**	90090
Genetic Algorithm	1160	2688	6784	17640548	108830
Evolution Strategy	1168	2654	6308	19600212	97880
Best known	**1150**	**2570**	6154	**17212548**	**88900**

6 ADVANTAGES/DISADVANTAGES

As with most heuristic algorithms there are inherent advantages and disadvantages. The most obvious disadvantage is that the optimal solution is not guaranteed to be found. However this is compensated for finding a very good solution in a practical amount of time. Another disadvantage is that this algorithm is based off of experimentation.

Perhaps the most significant advantage of the Ant Colony algorithms, versus other heuristic

algorithms, is that the Ant Colony algorithms work well with dynamic and "ill-structured"[3] problems. A great example of this, analyzed[4] in [Dhillon, 2006], is the Network Routing Problem.

7 SIGNIFICANCE OF [DORIGO, 1996]

With the biological inspired algorithm, [Dorigo, 1996] was able to make use of *positive feedback* as a search and optimization tool. Dorigo was also able to show that multiple agents (ants) acting together in cooperation is significantly better than the same number of agents (ants) acting independently of one another (as expected). The paper assures us that Ant Colony algorithms can be very beneficial and applied to many different problems due to its high-level ideas and ability to find good solutions to many very difficult combinatorial optimization problems.

As stated in [Dorigo, 1996], the main contributions of the paper are the following:

(i) The implementation of positive feedback as a search and optimization tool. That is, the key idea that if an ant makes a choice that turns out to be "good", then that choice will have be more desirable than before.

(ii) Synergy can show to be useful in distributed systems. The independently acting ants can cooperate to have a more effective search.

(iii) The useful applications of the Ant System to different combinatorial optimization problems.

REFERENCES

[Caro, 1998] G. Di Caro, M. Dorigo (1998) Ant Net: Distributed Stigmergetic Control for Communications Networks *Journal of Artificial Intelligence Research* 9(1998), 317 – 365.

[Dorigo, 1999] M. Dorigo, G. Di Caro, L. Gambardella (1999) Ant Algorithms for Discrete Optimization *Artificial Life*

[Dorigo, 1996] M. Dorigo, A. Colorni (1996) The Ant System: Optimization by a colony of cooperating agents *IEEE Transactions on Systems, Man, and Cybernetics-Part B* Vol.26, No.1, 1 – 13.

[Dorigo, 2004] M. Dorigo, T. Stutzle (2004) Ant Colony Optimization *MIT Press*

[Rizzoli, 2004] A.E. Rizzoli, F. Oliverio, R. Montemanni, L.M. Gambardella (2004) Ant Colony Optimisation for vehicle routing problems: from theory to applications *Publisher Name* ??(?), xx – yy.

[Blum, 2005] C. Blum (2005) Ant colony optimization: Introduction and recent trends *Physics of Life Reviews* 2, 353 – 373.

[Juenger, 2003] M. Juenger, G. Reinelt and G. Rinaldi (2003) Exact algorithms for NP-hard problems: A survey *Combinatorial Optimization - Eureka! You shrink!* 185 – 207

[3] "ill-structured" here means that perhaps the problem cannot be expressed concretely before the algorithm begins.

[4] Analysis of the AntNet algorithm in particular.

[Nugent, 1968] C.E. Nugent, T.E. Vollmann, J. Ruml (1968) An Experimental Comparison of Techniques for the Assignment of Facilities to Locations *Operations Research* 16, 150 – 173.

[Elshafei, 1977] Elshafei A.E. (1977) Hospital Layout as a Quadratic Assignment Problem *Operational Research Quarterly* 28, 167 – 179.

[Krarup, 1978] J. Krarup, P.M. Pruzan (1978) Computer-aided Layout Design *Mathematical Programing Study* 9, 85 – 94.

[Dhillon, 2006] S.S. Dhillion, P. Van Mieghem (2006) Performance analysis of the AntNet algorithm *Computer Networks* 51, 2104 – 2125.