Development of an automatically configurable ant colony optimization framework. State of art.

Aldar Saranov

March 14, 2017

Contents

Abstract

Some animal species show an extreme degree of social organization. Such species (e.g. ants) have pheromone production and detection body parts and therefore seize an ability to communicate between each other in indirect way. This concept has inspired the development of algorithms which are based on social behavior of population called ant colony optimization algorithms (ACO). These algorithms allow to solve NP-hard problems in a very efficient manner. Since these algorithms are considered metaheuristic the development of a ACO framework is the next step of formalizing of this area is to provide tools for resolving general optimization problems. This article gives the brief overview of the current ACO research area state, existing framework description and some tools which can be used for the framework automatic configuration.

1 Introduction (1 page)

Section descriptions. Pheromones. Constructive heuristics. Solution components. Problem models.

2 Combinatorial Optimization Problems and Constructive Heuristics (17 pages)

Combinatorial optimization problems (COP) are a whole class of mathematical optimization problems. These problems can be described by grouping, ordering, assigning or any other operations over the set of discrete objects. In practice one may need to resolve COP which have a large number of extra constraints for the solutions which are considered admissible. Many of these problems which are being thoroughly researched at the moment belong to NP-complete discrete optimization problems.

Definition

Optimization Problem is a tuple (Φ, ω, f), where

- Φ is a search space consisting of all possible assignments of discrete variables x_i, with $i = 1, ..., n$

- ω is a set of constraints for the decision variables
- $f : \Phi \rightarrow R$ is an objective function which has to be optimized

The problem describes the abstract subclass of tasks (e.g. find the minimum spanning tree of some graph) while the instance of a problem describes a certain practical problem (e.g. find the minimum spanning tree of a given graph G). The objective function in this case is the sum of the selected edges. One of the most frequently encountered problems is traveling salesman problem (TSP). Given a graph $G = (N, E)$ with $n = |N|$ nodes, where E - is a set of edges fully connecting the nodes and distances $d_{ij}, \forall(i,j) \in E$ one should find a Hamiltonian path of minimal length (in terms of sum of the weighted edges). The solution path can be represented as $\pi = (\pi_1, ..., \pi_n)^t$ of all n nodes, where pi_i is the node index at position i. The objective function is following

$$\min_{\pi \in \Phi} d_{\pi_i \pi_{i+1}} + \sum_{i+1}^{n-1} d_{\pi_i \pi_{i+1}} \tag{1}$$

TSP and QAP description. Solution components. Feasible solution. Permutation space.

3 The Concepts of Ant Colony Optimization

Algorithm

```
procedure ACO-Metaheuristic
    repeat
        for each ant do
            repeat
                ExtendPartialSolutionProbabilistically()
            until solution is complete
        for each ant ∈ SelectAntsForLocalSearch() do
            ApplyLocalSearch(ant)
        EvaporatePheromones()
        DepositPheromones()
    until termination criteria met
```

end

3.1 Choice of pheromone trails and heuristic information

C - Solution components.
$\tau_c \in T$ - pheromones of choosing.
$\tau'_c \in T'$ - pheromones of considering order.
π - candidate solution.
$\eta_c \in H$ - heuristic information (constant in time).

3.2 Solution construction

A solution is constructed by an ant.
Probabilistic rules:

- Classic
- Maniezzo
- Dorigo

α, β - choice parameters.
Extensions:

- Lookahead - pick several components at once[94]
- Candidate list - restriction of component choice at each step[33,34]
- Iterated greedy (partial deconstruction)[110]
- With external memory[1]
- Iterated ants[129]
- Cunning ants[128]
- Enhanced ACO[47]

3.3 Global pheromone update

Evaporation:
$\tau_{new} = evaporation(\tau_{old}, \rho, S^{eva})$
ρ - evaporation rate
S^{eva} - chosen solutions for evaporation

Deposition:
w_k - weight of solution s_k.
$F(S_k)$ - non-decreasing solution quality scaling function.

Update selection:

1. Ant system (update all)
2. Single update selections:
   (a) iteration-based update
   (b) global-based update
   (c) restart-based update

Update extensions:

1. Max-Min Ant System [122]
2. Rank-based Ant System [19]
3. Best-Worst Ant System [21]
4. Elitist Ant System [30, 36, 38]

3.4 Pheromone update schedule

Exploration vs exploitation.

3.5 Initialization of pheromones

3.6 Pheromone reinitialization

3.7 Local pheromone update

Parallel vs sequential. ACS [34]

3.8 Pheromone limits

MMAS and ACS examples.

3.9 Local search

Neighborhood operator. Best-improving and first-improving.

3.10 ACO algorithms as instantiations of the ACO Metaheuristc

4 Applications of ACO to other problem types

4.1 Continuous Optimization Problems

4.2 Multi-objective problems

4.3 Dynamic problems

4.4 Stochastic problems

5 ACO in combination with other methods

5.1 ACO and tree search methods

5.2 ACO and exact methods

5.3 ACO and surrogate models

5.4 Parameter adaptation

6 Existing ACO framework (5 pages)

6.1 Finding a better ACO configuration for the TSP

6.2 Finding a better ACO configuration for the QAP

7 IRACE automatic configuration (3 pages)

8 Conclusions

References