THE ANT COLONY OPTIMIZATION METAHEURISTIC

蚁群优化算法

Contents

Introduction

The inspiring source of ACO is the pheromone trail laying and following behavior of real ants which use pheromones as a communication medium.

ACO is based on the indirect communication of a colony of simple agents, called (artificial) ants, mediated by (artificial) pheromone trails.

The pheromone trails in ACO serve as a distributed, numerical information which the ants use to probabilistically construct solutions to the problem being solved and which the ants adapt during the algorithm's execution to reflect their search experience.

“

Introduction

Ant System (AS)

Ant Colony Optimization (ACO)

- static combinatorial optimization problems.
- dynamic combinatorial optimization problems.

TRADITIONAL APPROXIMATION APPROACHES

Construction Algorithms

procedure *Greedy Construction Heuristic*
    s_p = *empty solution*
    while s_p not_a_complete_solution **do**
        e = GreedyComponent(s_p)
        $s_p = s_p \otimes e$
    end
    return s_p
end *Greedy Construction Heuristic*

Local Search

procedure *IterativeImprovement* ($s \in \mathcal{S}$)
    s' = Improve(s)
    while $s' \neq s$ **do**
        $s = s'$
        s' = Improve(s)
    end
    return s
end *IterativeImprovement*

Problem Representation

problem

$P(S, f, \Omega)$

S *set of candidate solutions*
f the *objective function* which assigns to each candidate solution an objective function (cost) value $f(s,t)$
Ω a set of constraints.

components $C = \{c_1, c_2, \cdots, c_{N_C}\}$

The *states* of the problem $x = \langle c_i, c_j, \cdots, c_k, \cdots \rangle$

The set of all possible sequences χ

construction graph $G = (C, L)$

pheromone trail τ τ_{ij} c_i l_{ij}

heuristic value η

Ant Behaviors

- **It has a memory M.**
- **It can be assigned a start state and one or more termination conditions.**
- **If the ant can’t be moved to a feasible neighborhood, it might be allowed to move to a vertex j in its infeasible neighborhood**
- **It selects the move by applying a probabilistic decision rule.**
- **The construction procedure of ant k stops when at least one of the termination conditions is satisfied.**
- **When adding a component to the current solution it can update the pheromone trail associated to it or to the corresponding connection.**
- **Once built a solution, it can retrace the same path backward and update the pheromone trails of the used components or connections.**

“

The Metaheuristic

pheromone trail evaporation

- The pheromone deposited by previous ants decreases over time.
- It avoid a too rapid convergence of the algorithm towards a suboptimal region.
- It implements a useful form of forgetting, favoring the exploration of new areas of the search space.

daemon actions

- It implement centralized actions which cannot be performed by single ants.
- It choose to deposit extra pheromone on the components used by the ant that built the best solution by observing the path found by each ant in the colony(off-line pheromone updates).

“

Pseudo-code for ACO metaheuristic behavior

```
procedure ACO metaheuristic
    ScheduleActivities
        ManageAntsActivity()
        EvaporatePheromone()
        DaemonActions() {Optional}
    end ScheduleActivities
end ACO metaheuristic
```

The **ScheduleActivities** construct does not specify how these three activities are scheduled and synchronized. The designer is therefore free to specify the way these three procedures should interact.

Biological Analogy

In many ant species, individual ants may deposit a pheromone (a particular chemical that ants can smell) on the ground while walking .

By depositing pheromone they create a trail that is used, e.g., to mark the path from the nest to food sources and back.

In fact, by sensing pheromone trails foragers can follow the path to food discovered by other ants.

Also, they are capable of exploiting pheromone trails to choose the shortest among the available paths leading to the food.

“

Historical Development

AS

- *ant-cycle
- *ant-density*
- *ant-quantity*

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad \text{if } j \in \mathcal{N}_i^k$$

$$\eta_{ij} = 1/d_{ij}$$

$$\forall (i,j) \quad \tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k(t)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} 1/L^k(t) & \text{if edge } (i,j) \text{ is used by ant } k \\ 0 & \text{otherwise} \end{cases}$$

“

Improvement

elitist strategy

$$\Delta\tau_{ij}^{gb}(t) = \begin{cases} e/L^{gb}(t) & \text{if edge } (i,j) \in T^{gb} \\ 0 & \text{otherwise} \end{cases}$$

AS$_{\text{rank}}$

$$\forall (i,j) \quad \tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \sum_{r=1}^{w-1}(w-r)\cdot\Delta\tau_{ij}^{r}(t) + w\cdot\Delta\tau_{ij}^{gb}(t)$$

where $\Delta\tau_{ij}^{r}(t) = 1/L^{r}(t)$ and $\Delta\tau_{ij}^{gb}(t) = 1/L^{gb}$.

Ant Colony System (ACS)

$$q_0, 0 \le q_0 \le 1, \quad j = \arg\max_{j\in\mathcal{N}_i^k}\{\tau_{ij}(t)\cdot\eta_{ij}^{\beta}\},$$

$$1-q_0 \qquad p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{l\in\mathcal{N}_i^k}[\tau_{il}(t)]^{\alpha}\cdot[\eta_{il}]^{\beta}} \quad \text{if } j \in \mathcal{N}_i^k$$

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\Delta\tau_{ij}^{\text{best}}(t)$$

$\mathcal{MAX}$–$\mathcal{MIN}$ Ant System

$$\tau_{\min} \le \tau_{ij} \le \tau_{\max} \; \forall\tau_{ij},$$

Applications

NP-hard problems: For these problems, very often ACO algorithms are coupled with extra capabilities such as problem-specific local optimizers, which take the ants' solutions to local optima.

Shortest path problems: in which the properties of the problem's graph representation change over time concurrently with the optimization process that has to adapt to the problem's dynamics. In this case we conjecture that the use of ACO algorithms becomes more and more appropriate as the variation rate of the costs increases and/or the knowledge about the variation process diminishes.

APPLICATION PRINCIPLES

- Pheromone Trails Definition
- *Balancing Exploration and Exploitation*

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}} \quad \text{if } j \in \mathcal{N}_i^k$$

- *ACO and Local Search*

ACO algorithms perform best when coupled with local search algorithms

- *Heuristic Information*
- *Number of Ants*
- *Candidate Lists*

conclusions

- the great majority of problems attacked by ACO are static and welldefined combinatorial optimization problems
- the role played by local search is extremely important to obtain good results
- The writer believes that ACO algorithms will really show their greatest advantage when they are systematically applied to "ill–structured" problems for which it is not clear how to apply local search, or to highly dynamic domains with only local information available.