# Ant colony optimization theory

Wojciech Grabis

January 15, 2020

# Quick Introduction

- Inspired by foraging behaviour of ants.
- Ants guide other ants by leaving pheromone trail.
- Quantity of pheromone depends on quality and quantity of food.

# Overview

1. Ant colony optimization
2. Convergence of ACO algorithms
3. Model-based search
4. Search bias in ACO algorithms

# Problem definition

Hard combinatorial optimization (CO) problem definition.

- A model $P = (S, \Omega, f)$ consisting of:
  - a search (or solution) space S defined over a finite set of discrete variables and a set $\Omega$ of constraints among the variables.
  - an objective function $f : S \to \mathbb{R}^+$ to be minimized.
- Search space S is a set of *n discrete variables* $X_i$ with values $v_i^j \in D_i = \{v_i^1, \dots v_i^{|D_i|}\}, i = 1, \dots, n$. Variable instantiation is the assignment $X_i = v_j^i$.
- A feasible solution $s \in S$ is a complete assignment that satisgies the constraints $\Omega$. If $\Omega = \emptyset$, then each variable can take any value from its domain, and we call $P$ an *unconstrained* problem model.
- A feasible solution $s^* \in S$ is called a *global optimum*, if $f(s^*) \leqslant f(s) \forall s \in S$.

# Pheromone trail

- We call a combination of a decision variable $X_i$ and one of its domain values $v_i^j$ a *solution component* denoted as $c_i^j$. Set of all solution components is denoted as $\mathfrak{C}$.
- Pheromone model consists of a *pheromone trail parameter* $\mathcal{T}_i^j$ for each solution component $c_i^j$. The value of pheromone trail parameter $\mathcal{T}_i^j$ called *pheromone value* is denoted by $\tau_i^j$.

# Example ATSP

Example of a model for asymmetric traveling salesman problem:

- We are given a graph $G(V, E)$ with a positive weight $d_{ij}$ for each edge $e_{ij} \in A$. The goal is to find the shortest Hamiltonian cycle in $G$.
- We model each city $i \in V$ by a decision variable $X_i$, with a domain value $v_i^j$ for each outgoing edge $e_{ij}$.
- The set of constraints so that, only those solutions that form a Hamiltonian cycle are valid.
- For each variable $X_i$ and value $v_i^j$ we have solution component $c_i^j$ with which we build our set $\mathfrak{C}$.
- Lastly we build pheromone model $\mathcal{T}$, by associating a pheromone trail parameter $\mathcal{T}_i^j$ with value $\tau_i^j$ for each solution component $c_i^j$.

# Framework of a basic ACO algorithm

**Algorithm 1** The framework of a basic ACO algorithm

```
input: An instance P of a CO problem model 𝒫 = (𝒮, f, Ω).
InitializePheromoneValues(𝒯)
𝔰_bs ← NULL
while termination conditions not met do
    𝔖_iter ← ∅
    for j = 1, ..., n_a do
        𝔰 ← ConstructSolution(𝒯)
        if 𝔰 is a valid solution then
            𝔰 ← LocalSearch(𝔰)        {optional}
            if (f(𝔰) < f(𝔰_bs)) or (𝔰_bs = NULL) then 𝔰_bs ← 𝔰
            𝔖_iter ← 𝔖_iter ∪ {𝔰}
        end if
    end for
    ApplyPheromoneUpdate(𝒯,𝔖_iter,𝔰_bs)
end while
output: The best-so-far solution 𝔰_bs
```

# Framework ACO components

- **InitializePheromoneValues**($\mathcal{T}$): at the start initialize all pheromone values with a constant $c > 0$.
- **ConstructSolution**($\mathcal{T}$): constructive heuristic for pobabistically constructing solutions.

# ConstructSolution

The algorithm of constructing a solution:

- Construction starts with an empty partial solution $s^p = \emptyset$.
- At each step $s^p$ is extended by a feasible solution component from set $\mathfrak{R}(s^p) \subseteq \mathfrak{C} \setminus \{s^p\}$. By feasible we mean solution components that meet the constraints.

# ConstructSolution cont.

- Choice of a solution component $c_i^j \in \mathfrak{R}(s^p)$ is done with respect to the pheromone model. The probability of $c_i^j$ is proportional to $[\tau_i^j]^\alpha * [n(c_i^j)]^\beta$, where $n$ is a function that assigns additional information to each solution component called *heuristic information*.
- Parameters $\alpha > 0$ and $\beta \geqslant 0$ are used to determine the value of pheromone and heuristic information. Heuristic information is an optional function.
- Most ACO algorithms define the probability of choosing next solutioncomponent, called *transition probability*, as follows:

$$p(c_i^j|s^p) = \frac{[\tau_i^j]^\alpha * [n(c_i^j)]^\beta}{\sum_{c_k^l \in R(s^p)} [\tau_k^l]^\alpha * [n(c_k^l)]^\beta}, \forall c_i^j \in \mathfrak{R}(s^p)$$

# Framework ACO components

- **LocalSearch**$(s)$: local search procedure, that can be optionally used to improve solution generated by ants.
- **ApplyPheromoneUpdate**$(\mathcal{T}, \mathfrak{S}_{iter}, s_{bs})$: pheromone update rule, that is used to increase the pheromone values on solution components that have been found in high quality solutions.

# ApplyPheromoneUpdate

Most ACO algorithms use a variation of:

$$\tau_i^j = (1-p) * \tau_i^j + \frac{p}{\mathfrak{S}_{upd}} * \sum_{\{s \in \mathfrak{S}_{upd} | c_i^j \in s\}} F(s),$$

$$\forall i = 1, \ldots, n, j = 1, \ldots, |D_i|$$

- Set $\mathfrak{S}_{upd}$ depends on specification of algorithm, $\mathfrak{S}_{upd} \subseteq \mathfrak{S}_iter \cup \{s_{bs}\}$.
- Parameter $p \in (0, 1]$ called *evaporation rate* is used to decrease all pheromone values.
- Function $F : \mathfrak{S} \to \mathbb{R}^+$ is called *quality function*, such that $f(s) < f(s') \implies +\infty > F(s) \geq F(s'), \forall s \neq s' \in \mathfrak{S}$.

# ACO variants

Variants of ACO algorithms usually differ in the pheromone update rule used. Main examples:

- Ant System (AS) update rule:

$$\mathfrak{S}_{upd} \leftarrow \mathfrak{S}_{iter}$$

- Iteration Based (IB) update rule:

$$\mathfrak{S}_{upd} \leftarrow argmax\{F(s)|s \in \mathfrak{S}_{iter}\}$$

- Best-so-far (BS) update rule:

$$\mathfrak{S}_{upd} \leftarrow \{s_{bs}\}$$

# Upper bound

## Proposition 1

*Given ACO algorithm using defined pheromone update rule, for any pheromone value $\tau_i^j$, the following holds:*

$$\lim_{x \to \infty} \tau_i^j \leq \frac{F(s^*) * |\{\mathfrak{S}\}|}{p}$$

# The hyper-cube framework

Hyper-cube framework(HCF) for ACO:

- We regard the vector of phermone values as a $|\mathfrak{C}|$-dimensional vector $\vec{\tau}$.
- The application of a pheromone value update moves it in a $|\mathfrak{C}|$-dimensional hyper-space.
- The pheromone update rule in HCF-form becomes:

$$\tau_i^j \leftarrow (1-p) * \tau_i^j + p * \sum_{\{s \in \mathfrak{S}_{upd} | c_i^j \in s\}} \frac{F(s)}{\sum_{\{s' \in \mathfrak{S}_{upd}\}} F(s')}$$

# The hyper-cube framework cont.

Graphical interpretation of HCF:

- We can interpret $s \in \mathfrak{S}$ as a binary vector $\vec{s}$ of dimension $|\mathfrak{C}|$, in which components $c_i^j$ that belong to the solution are set 1, and other to 0. That means the solution $s$ is a corner of the $|\mathfrak{C}|$-dimensional unit hyper-cube.
- We can denote the update rule of pheromone value as: $\vec{\tau} \leftarrow (1-p) * \vec{\tau} + p * \vec{m}$, where $\vec{m} = \sum_{s \in \mathfrak{S}_{upd}} \frac{F(s)}{\sum_{s' \in \mathfrak{S}_{upd}} F(s')} * \vec{s}$

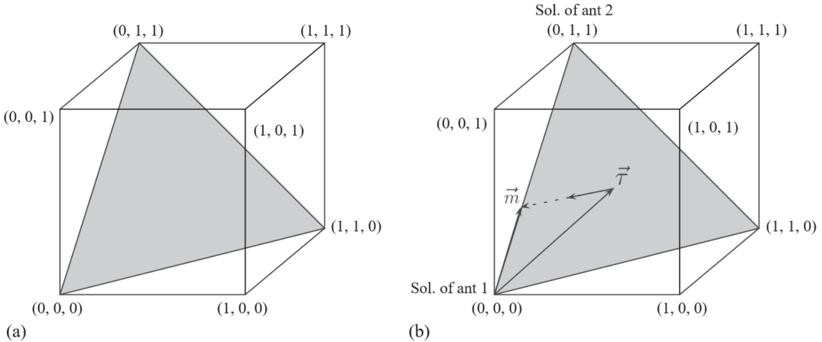
# The hyper-cube framework cont.

- Additionally if we denote the convex hull of $\mathfrak{C}$ by $conv(\mathfrak{C})$, it holds that

$$\vec{\tau} \in conv(\mathfrak{C}) \Leftrightarrow \vec{\tau} \ = \sum_{s \in \mathfrak{C}} \alpha_s \vec{s}, \qquad \alpha_s \in [0,1], \quad \sum_{s \in \mathfrak{C}} \alpha_s = 1$$

- This shows that the application of the pheromone update rule in the HCF shifts the pheromone value vector $\vec{\tau}$ toward $\vec{m}$. And if initial pheromone value vector $\vec{\tau} \in conv(\mathfrak{C})$, it remains in $conv(\mathfrak{C})$.

# The hyper-cube framework example



# Models of ACO algorithms

Model of an ACO algorithm is a deterministic dynamical system obtained by applying the expected pheromone update instead of the real pheromone update.

Model can be used to study the behaviour of algorithm, eg. by studying the iteration quality.

Models are defined as follows:

$$M(problem, update\_rule, number\_of\_ants),$$

For example: $M(*, AS, n_a < \infty)$ or $M(*, AS, n_a = \infty)$.

# Iteration quality

One of the main properties of ACO algorithms theoretical properties is the iteration quality of a model. Denoted as $W_F(\mathcal{T})$ or $W_F(\mathcal{T}|t)$, where $t > 0$ is the iteration counter.

For example for model $M(*, AS, n_a = \infty)$, we have iterative quality given by:

$$W_F(\mathcal{T}) = \sum_{s \in \mathfrak{S}} F(s) * p(s|\mathcal{T})$$

and expected pheromone update by:

$$\tau_i^j \leftarrow (1 - p) * \tau_i^j + p * \sum_{\{s \in \mathfrak{S} | c_i^j \in s\}} F(s) * p(s|\mathcal{T})$$

# Convergence of ACO algorithms

- *Convergence in value* - the probability that the algorithm will generate an optimal solution at least once if given enough time.
- *Convergence in solution* - the probability that the algorithm reaches a state which keeps generating the same optimal solution.

Paper focuses on convergence of two ACO algorithms classes: $ACO_{bs,\tau_{min}}$ and $ACO_{bs,\tau_{min}(t)}$, using simplified transition probability:

$$p(c_i^j = \frac{[\tau_i^j]^\alpha}{\sum_{c_k^l \in R(s^p)} [\tau_k^l]^\beta}, \forall c_i^j \in \mathfrak{R}(s^p)$$

# Convergence of value $ACO_{bs,\tau_{min}}$

## Proposition 2

*Once an optimal solution $s^*$ has benn found by algorithm $ACO_{bs,\tau_{min}}$, it holds that:*

$$\forall c_i^j \in s^* : \lim_{t\to\infty} \tau_i^j(t) = \tau_{max} = \frac{F(s^*)}{p}$$

## Theorem 1

*Let $\mathbf{p}^*(t)$ be the probability that $ACO_{bs,\tau_{min}}$ finds an optimal solution at least once within the first t iterations. Then, for an arbitrarily small $\epsilon > 0$ and for sufficicently large t it holds that*

$$\boldsymbol{p}^*(t) \geq 1 - \epsilon$$

*and asymptotically* $\lim_{t\to\infty} \boldsymbol{p}^*(t) = 1$.

# Convergence of value $ACO_{bs,\tau_{min}(t)}$

**Theorem 2**

*Let the lower pheromone trail limits in $ACO_{bs,\tau_{min}(t)}$ be*

$$\forall t \geq 1, \tau_{min}(t) = \frac{d}{ln(t+1)}$$

*with d being a constant, and let $\boldsymbol{p}^*(t)$ be the probability that $ACO_{bs,\tau_{min}(t)}$ finds an optimal solution at least once within the first t iterations. Then it holds that*

$$\lim_{t \to \infty} \boldsymbol{p}^*(t) = 1$$

# Convergence of solution $ACO_{bs,\tau_{min}(t)}$

**Theorem 3**

*Let $t^*$ be the iteration in which the first optimal solution $s^*$ has been found and $\boldsymbol{p}(s^*, t, k)$ be the probability that an arbitrary ant $k$ constructs $s^*$ in the $t$-th iteration, with $t > t^*$. Then it holds that*

$$\lim_{t \to \infty} \boldsymbol{p}(s^*, t, k) = 1$$

# Convergence summary

### Open problem 1

*The proofs of convergence do not say anything about the time required to find an optimal solution, which can be astronomically large. It would be interesting to obtain results on convergence speed for ACO algorithms, in spirit similar to what has been done in evolutionary computation for relatively simple problems such as for example, ONE-MAX.*

# Model-based search

Model-base search is an algorithmic framework, that uses a parametrized probabilistic model to generate solutions to the problem under consideration. The optimization problem is replaced by following continous maximization problem:

$$\tau^* \leftarrow argmax_{\tau} \; W_f(\mathcal{T})$$

# Stochastic gradient ascent

To calculate the optimum of previous maximization problem we can use stochastic gradient ascent method,as a heuristic to change $\tau$ in each iteration. The calculation of next value is as follows:

$$\tau(t+1) = \tau(t) + \alpha \sum_{s \in \mathfrak{S}_{upd}} F(s) \nabla ln \boldsymbol{p}(s|\tau(t))$$

.

# Search bias in ACO algorithms

- Positive - directs the search towards good zones of search space (pheromone update rule).
- Negative - caused by e.g. pheromone model or the solution construction process.
  - Negative search bias caused by unfair competition.
  - Search bias caused by selection fix-points.

# Unfair competition bias - unconstrained problems

**Theorem 4**

*The expected iteration quality $W_F(\mathcal{T})$ of $M(U, HCF - AS, n_a = \infty)$, where $U$ stands for the application to unconstrained problems, is continuously non-decreasing. More formally it holds that*

$$W_F((T|t+1) > W_F(\mathcal{T}|t)$$

*as long as at least one pheromone value changes from iteration $t$ to iteration $t+1$.*

# Unfair competition - constrained problems

Paper illustrates this bias using k-MST problem.

Given an undirected graph $G = (V, E)$, where $|V| = n$ and $|E| = m$, with edge-weights $w(e) \in \mathbb{N}^+, \forall e \in E$. The goal is to find a tree $T_k$ that minimizes:

$$f(T_k) = \sum_{e \in E(T_k)} w(e)$$

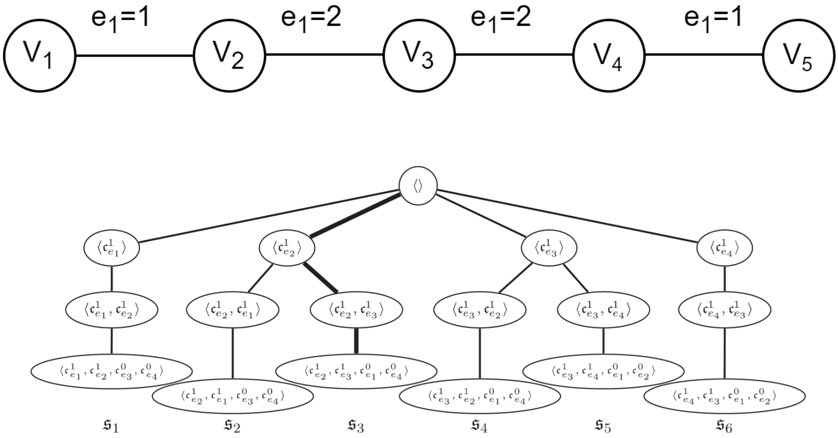



Figure: *Possible solutions for our instance.*

# k-MST example

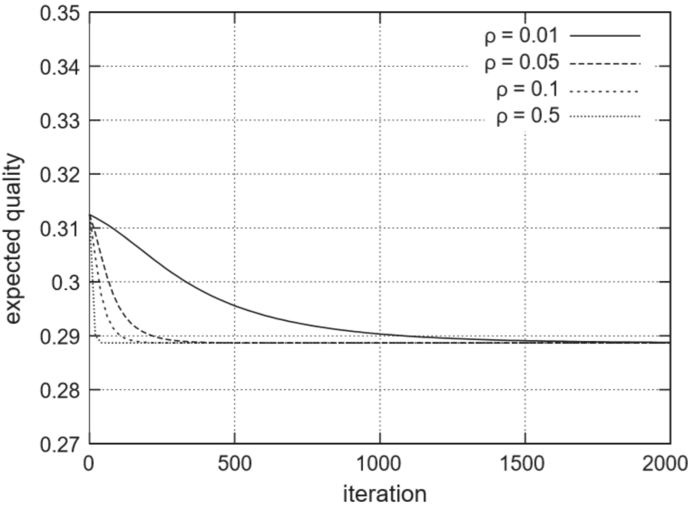



Figure: *Evolution of the expected iteration quality $W_F$ of model $M(U, IB, n_a = \infty$.*

# k-MST example

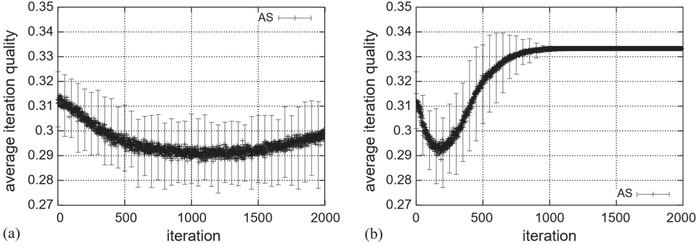



Figure: *Evolution of average iteration quality obtained by AS algorithm for* $n_a = 10$

# Competition balanced system

## Definition 1

*Given a model $\mathcal{P}$ of a CO problem, we call the combination of an ACO algorithm and a problem instance $P$ of $\mathcal{P}$ a competition-balanced-system, if the following holds: given a feasible partial solution $s^p$ and the set of solution components $\mathfrak{N}(s^p)$ that can be added to extend $s^p$, each solution component $c \in \mathfrak{N}(s^p)$ is a component of the same number of feasible solutions as any other solution component $c' \neq c$.*

## Open problem 2

*Is the property of being a competition-balanced system sufficent to ensure the absence of any negative search bias?*

# Temporary quality decrease

**Open problem 3**

*Can it be shown that the models $M(U, IB, n_a = \infty)$ or $(U, BS, n_a = \infty)$ have stable attractors that correspond to solutions to the problems.*

# Selection fix points

What if we use 1 ant in our ACO algorithm?

- In case of example model $M(U, AS, n_a = 1)$, we have

$$\tau_i^j(t+1) = (1-p)*\tau_i^j(t) + p*\delta_i^j, \; where \; \delta_i^j = \boldsymbol{p}(c_i^j|\mathcal{T}) = \frac{\tau_i^j(t)}{\sum\limits_{k=1,\ldots,|D_i|}} \tau_i^k(t)$$

- This makes $\tau_i^j$, move towards $\delta_i^j$.
- Those situations were labeled as *selection fix points*

# Conclusions