Ant Colony Optimization

Christian Blum

ALBCOM, LSI,
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONA, SPAIN

Overview

What is ant colony optimization (ACO)? A technique for optimization whose inspiration is the foraging behaviour of real ant colonies.

Different aspects:

- **Swarm intelligence:** Origins and inspiration of ACO
- **The ACO metaheuristic:**
  - How does it work?
  - Examples
  - Hybridization with other optimization techniques
- **Theoretical studies:**
  - Model-based search: Similarities to other algorithms
  - Negative search bias

Swarm intelligence

The origins of ant colony optimization

Swarm intelligence

Inspiration: Collective behaviour of social insects, flocks of birds, or fish schools

Examples of social insects:

- Ants
- Termites
- Some wasps and bees

Some facts:

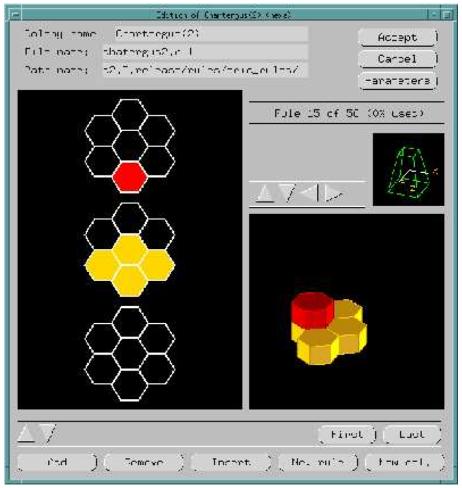
- About 2% of all insects are social
- About 50% of all social insects are ants
- Total weight of ants is about the total weight of humans
- Ants colonize world since 100.000.000 years, humans only since 50.000 years

Swarm intelligence

Natural examples

- Nest building behaviour of wasps
- Agricultural behaviour of leaf-cutter ants
- Nest building behaviour of weaver ants
- Cemetary building behaviour of ants
- Foraging behaviour of ants

Swarm intelligence

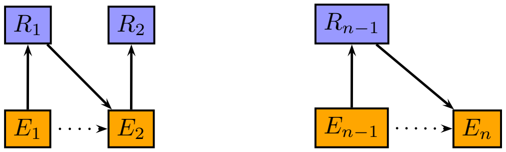


© Dominique Snyers (http://www-iasc.enst-bretagne.fr/PROJECTS/SWARM/nest.html)

Swarm intelligence

Properties of social societies:

- Consist of a set of simple entities
- Distributedness: No global control
- Self-organization by:
  - **Direct communication:** visual, or chemical contact
  - **Indirect communication:** Stigmergy (Grassé, 1959)



Result: Complex tasks can be accomplished in cooperation

Swarm intelligence

© Dominique Snyers (http://www-iasc.enst-bretagne.fr/PROJECTS/SWARM/nest.html)

Swarm intelligence

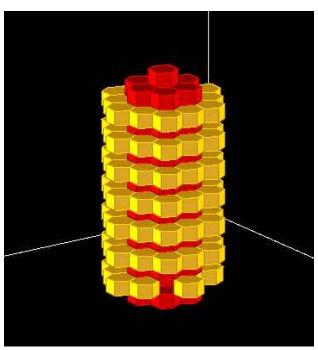
Natural examples

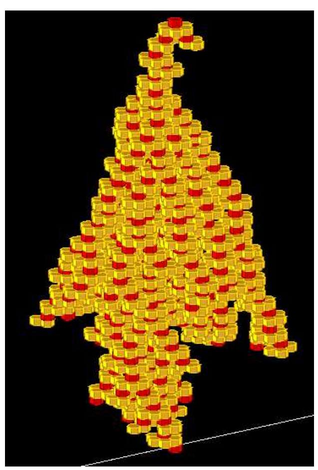
- Nest building behaviour of wasps
- Agricultural behaviour of leaf-cutter ants
- Nest building behaviour of weaver ants
- Cemetary building behaviour of ants
- Foraging behaviour of ants

Swarm intelligence

© Alex Wild (http://www.myrmecos.net)

Swarm intelligence





© Dominique Snyers (http://www-iasc.enst-bretagne.fr/PROJECTS/SWARM/nest.html)

Swarm intelligence

© Alex Wild (http://www.myrmecos.net)

Swarm intelligence

Natural examples

- Nest building behaviour of wasps
- Agricultural behaviour of leaf-cutter ants
- Nest building behaviour of weaver ants
- Cemetary building behaviour of ants
- Foraging behaviour of ants

Swarm intelligence

Swarm intelligence

Swarm intelligence

Swarm intelligence

Natural examples

- Nest building behaviour of wasps
- Agricultural behaviour of leaf-cutter ants
- Nest building behaviour of weaver ants
- Cemetary building behaviour of ants
- Foraging behaviour of ants

Ant colony optimization, June 26, 2005, GECCO 2005, Washington, US © C. Blum

Swarm intelligence

Natural examples

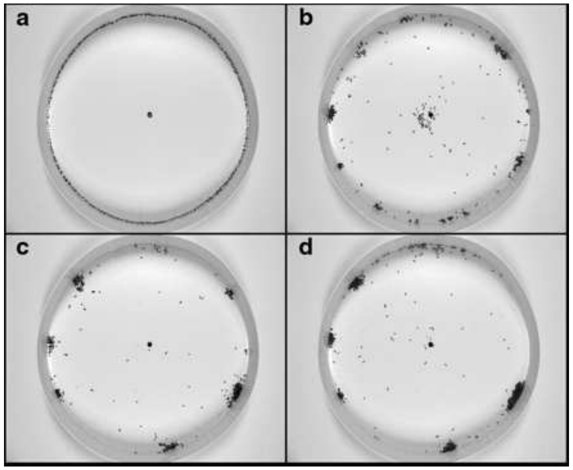
- Nest building behaviour of wasps
- Agricultural behaviour of leaf-cutter ants
- Nest building behaviour of weaver ants
- Cemetary building behaviour of ants
- Foraging behaviour of ants

Ant colony optimization, June 26, 2005, GECCO 2005, Washington, US © C. Blum

Swarm intelligence

Ant colony optimization, June 26, 2005, GECCO 2005, Washington, US © C. Blum

Swarm intelligence



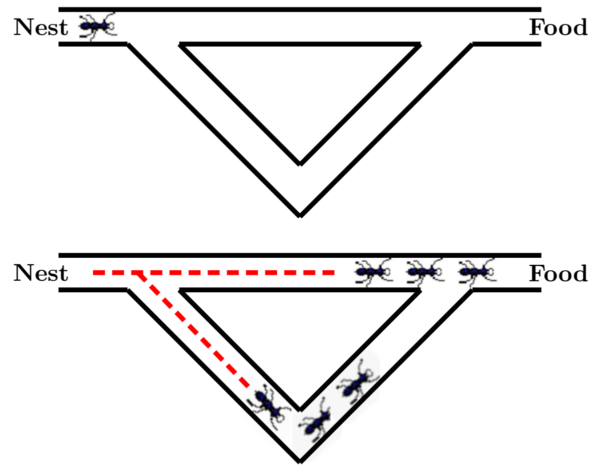
Swarm intelligence

Communication strategies:

- Direct communication: For example, recruitment
- Indirect communication: via chemical pheromone trails

Basic behaviour:

Swarm intelligence



Swarm intelligence

Communication strategies:

- Direct communication: For example, recruitment
- Indirect communication: via chemical pheromone trails

© Alex Wild (http://www.myrmecos.net)

© Christian Blum

Swarm intelligence

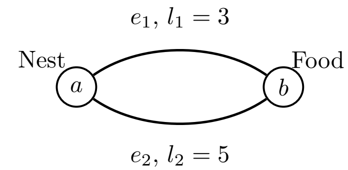
© Alex Wild (http://www.myrmecos.net)

The ant colony optimization metaheuristic

- Foraging behaviour → technical algorithm
- Example: traveling salesman problem
- Closer look at algorithm components
- Hybridization with other techniques

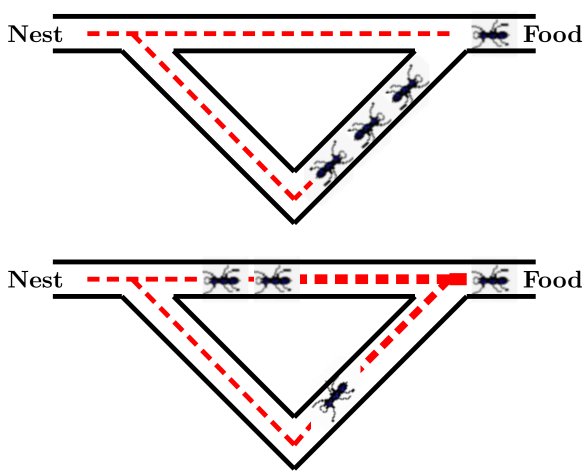
The ant colony optimization metaheuristic

Technical simulation:



- Introduce parameters $\mathcal{T}_1$ for e_1, and $\mathcal{T}_2$ for e_2
- Initialize the parameter values $\tau_1 = \tau_2 = c > 0$, → artificial pheromone

Swarm intelligence

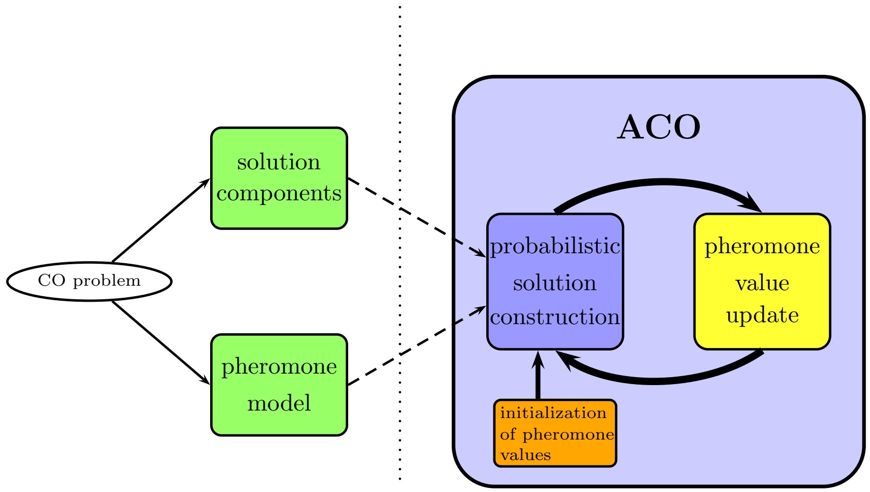


The ant colony optimization metaheuristic

Metaheuristics:

- Simulated Annealing (SA) [Kirkpatrick, 1983]
- Tabu Search (TS) [Glover, 1986]
- Genetic and Evolutionary Computation (EC) [Goldberg, 1989]
- **Ant Colony Optimization (ACO) [Dorigo, 1992]**
- Greedy Randomized Adaptive Search Procedure (GRASP) [Resende, 1995]
- Guided Local Search (GLS) [Voudouris, 1997]
- Iterated Local Search (ILS) [Stützle, 1999]
- Variable Neighborhood Search (VNS) [Mladenović, 1999]

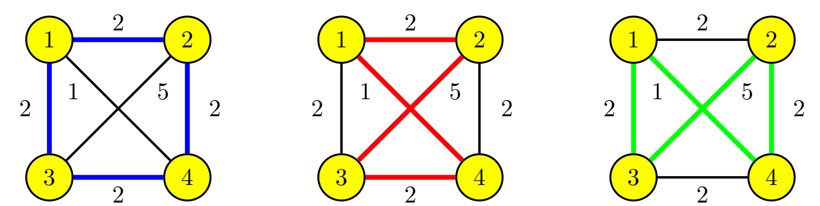
The ant colony optimization metaheuristic



The ant colony optimization metaheuristic

TSP in terms of a combinatorial optimization problem $\mathcal{P} = (\mathcal{S}, f)$:

- $\mathcal{S}$ consists of all possible Hamiltonian cycles in G.
- Objetive function $f : \mathcal{S} \mapsto I\!R^+$: $s \in \mathcal{S}$ is defined as the sum of the edge-weights of the edges that are in s.



obj. function value: 8 obj. function value: 10 obj. function value: 10

The ant colony optimization metaheuristic

Algorithm:

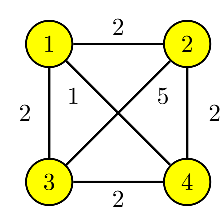
Iterate:

- Place n_a ants in node a.
- Each of the n_a ants traverses from a to b either
  - via e_1 with probability $\mathbf{p}_1 = \frac{\tau_1}{\tau_1+\tau_2}$,
  - or via e_2 with probability $\mathbf{p}_2 = 1 - \mathbf{p}_1$.
- Evaporate the artificial pheromone: $\tau_i \leftarrow \rho\tau_i$, $i = 1, 2$, $\rho \in (0, 1]$
- Each ant leaves pheromone on its traversed edge e_i: $\tau_i \leftarrow \tau_i + \frac{1}{l_i}$

Problem: In combinatorial optimization → exp. # of solutions

The ant colony optimization metaheuristic

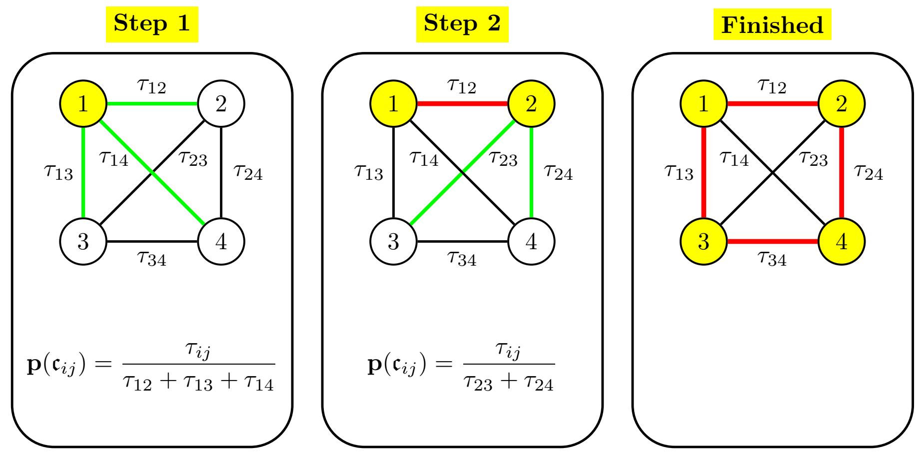
Example: Travelling salesman problem (TSP). Given a completely connected, undirected graph $G = (V, E)$ with edge-weights.



Goal: Find a tour (a Hamiltonian cycle) in G with minimal sum of edge weights.

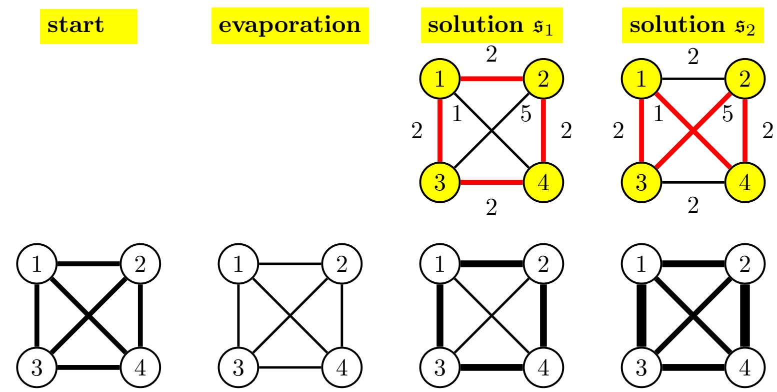
The ant colony optimization metaheuristic

TSP example: Tour construction



The ant colony optimization metaheuristic

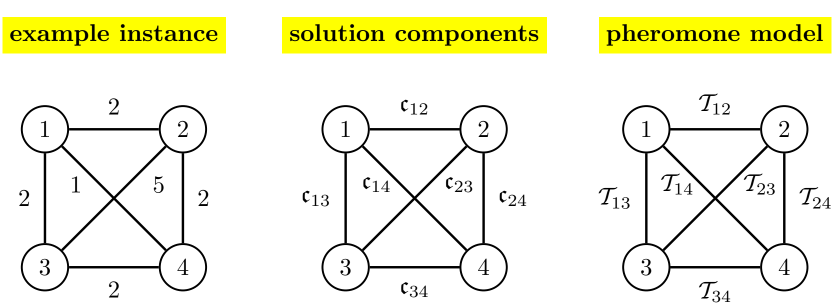
TSP example: Updating the pheromone values using the AS-update rule



The ant colony optimization metaheuristic

Example: Travelling salesman problem (TSP)

Derivation of solution components and pheromone model



The ant colony optimization metaheuristic

TSP example: Updating the pheromone values using the AS-update rule

Pheromone evaporation

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij}$$

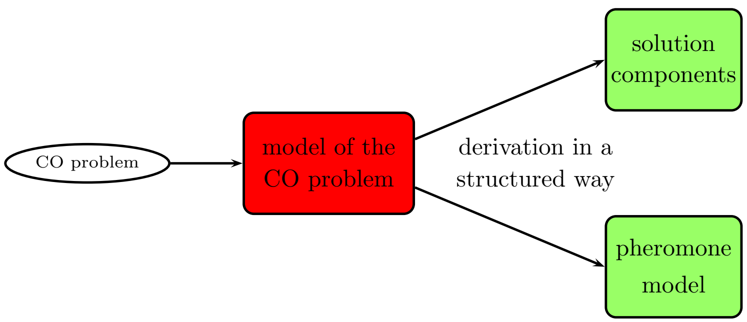
Reinforcement

$$\tau_{ij} \leftarrow \tau_{ij} + \rho \cdot \sum_{\{\mathfrak{s}\in\mathfrak{S}_{iter} \mid \mathfrak{c}_{ij}\in\mathfrak{s}\}} F(\mathfrak{s})$$

where

- evaporation rate $\rho \in (0,1]$
- $\mathfrak{S}_{iter}$ is the set of solutions generated in the current iteration
- quality function $F : \mathfrak{S} \mapsto \mathbb{R}^+$. We use $F(\cdot) = \frac{1}{f(\cdot)}$

The ant colony optimization metaheuristic



The ant colony optimization metaheuristic

Deriving the generic solution components:

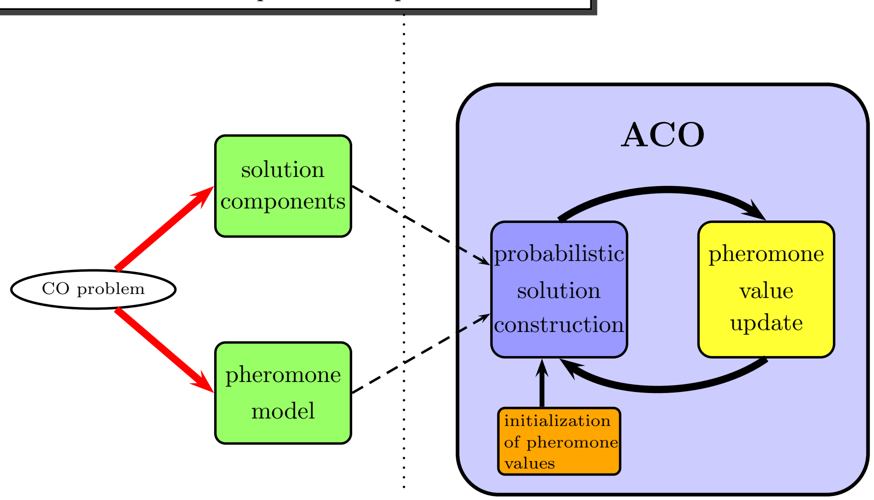
For each combination of X_i and $v_i^j \in D_i = \{v_i^1, \ldots, v_i^{|D_i|}\}$ $\Rightarrow$ generic solution component $\mathfrak{c}_i^j \in \mathfrak{C}$

Deriving the pheromone model:

For each $\mathfrak{c}_i^j \in \mathfrak{C}$ $\Rightarrow$ pheromone trail parameter $\mathcal{T}_i^j \in \mathcal{T}$

The ant colony optimization metaheuristic

Definition of solution components and pheromone model



The ant colony optimization metaheuristic

Definition: CO problem model

A *model* $\mathcal{P} = (\mathcal{S}, \Omega, f)$ of a CO problem consists of:

- a search (or solution) space $\mathcal{S}$ defined over
  - a finite set of n discrete decision variables X_i $(i = 1, \ldots, n)$;
  - and a set Ω of *constraints* among the variables;
- an *objective function* $f : \mathcal{S} \rightarrow I\!R^+$ to be minimized.

If the set of constraints Ω is empty, $\mathcal{P}$ is an unconstrained problem model, otherwise a constrained problem model.

The ant colony optimization metaheuristic

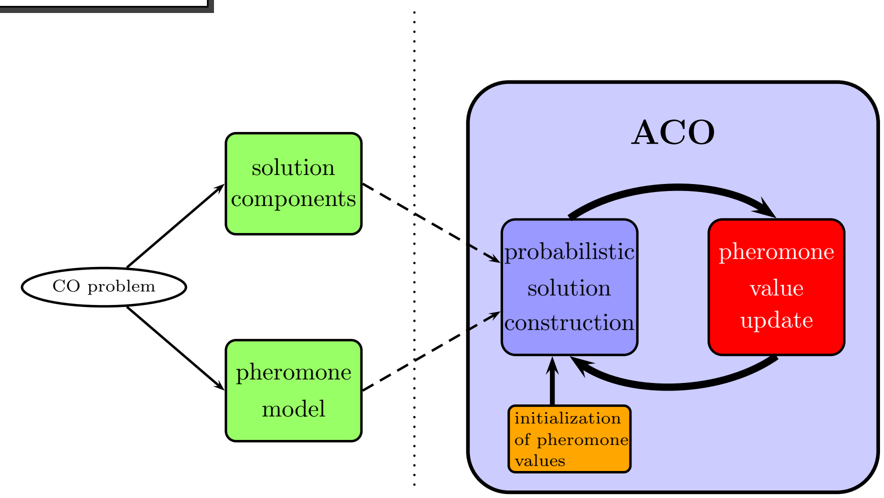
A general constructive heuristic:

- $\mathfrak{s}^p = \langle\rangle$
- Determine $\mathfrak{N}(\mathfrak{s}^p)$
- **while** $\mathfrak{N}(\mathfrak{s}^p) \neq \emptyset$
  - $\mathfrak{c} \leftarrow \mathsf{ChooseFrom}(\mathfrak{N}(\mathfrak{s}^p))$
  - $\mathfrak{s}^p \leftarrow$ extend $\mathfrak{s}^p$ by adding solution component $\mathfrak{c}$
  - Determine $\mathfrak{N}(\mathfrak{s}^p)$
- **end while**

Problem: How to implement function $\mathsf{ChooseFrom}(\mathfrak{N}(\mathfrak{s}^p))$?

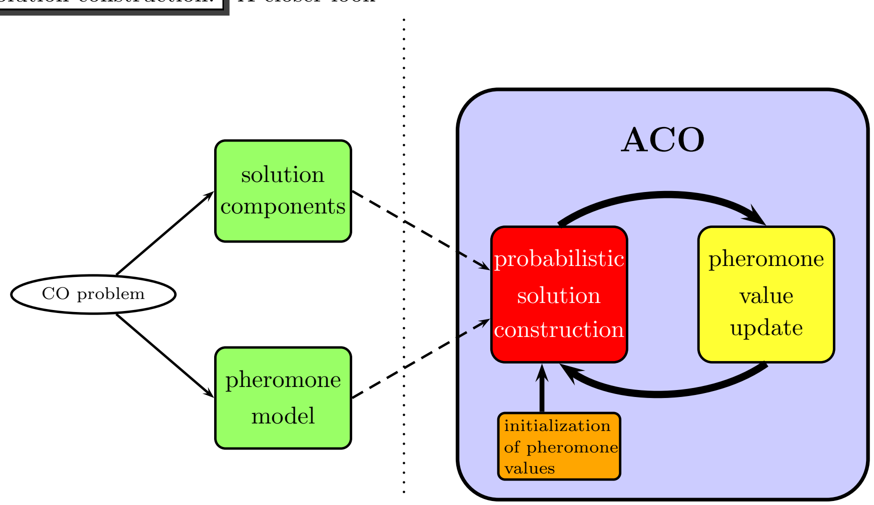
The ant colony optimization metaheuristic

Pheromone update: A closer look



The ant colony optimization metaheuristic

Solution construction: A closer look



The ant colony optimization metaheuristic

Possibilities for implementing $\mathsf{ChooseFrom}(\mathfrak{N}(\mathfrak{s}^p))$:

- Greedy algorithms:

$$\mathfrak{c}^* = \mathrm{argmax}_{\mathfrak{c}_i^j \in \mathfrak{N}(\mathfrak{s}^p)} \eta(\mathfrak{c}_i^j) \ ,$$

  where $\eta : \mathfrak{C} \mapsto I\!R^+$ is a Greedy function

- Ant colony optimization:

$$\mathbf{p}(\mathfrak{c}_i^j \mid \mathfrak{s}^p) = \frac{[\tau_i^j]^\alpha \cdot [\eta(\mathfrak{c}_i^j)]^\beta}{\sum\limits_{\mathfrak{c}_k^l \in \mathfrak{N}(\mathfrak{s}^p)} [\tau_k^l]^\alpha \cdot [\eta(\mathfrak{c}_k^l)]^\beta} \ , \quad \forall\, \mathfrak{c}_i^j \in \mathfrak{N}(\mathfrak{s}^p) \ ,$$

  where α and β are positive values

Observation: ACO can be applied if a constructive heuristic exists!

The ant colony optimization metaheuristic

ACO update variants:

AS-update	$\mathfrak{S}_{upd} \leftarrow \mathfrak{S}_{iter}$ weights: $w_{\mathfrak{s}} = 1 \; \forall \; \mathfrak{s} \in \mathfrak{S}_{upd}$
elitist AS-update	$\mathfrak{S}_{upd} \leftarrow \mathfrak{S}_{iter} \cup \{\mathfrak{s}_{bs}\}$ ($\mathfrak{s}_{bs}$ is best found solution) weights: $w_{\mathfrak{s}} = 1 \; \forall \; \mathfrak{s} \in \mathfrak{S}_{iter}$, $w_{\mathfrak{s}_{bs}} = e \geq 1$
rank-based AS-update	$\mathfrak{S}_{upd} \leftarrow$ best $m-1$ solutions of $\mathfrak{S}_{iter} \cup \{\mathfrak{s}_{bs}\}$ (ranked) weights: $w_{\mathfrak{s}} = m - r$ for solutions from $\mathfrak{S}_{iter}$, $w_{\mathfrak{s}_{bs}} = m$
IB-update:	$\mathfrak{S}_{upd} \leftarrow \operatorname{argmax}\{F(\mathfrak{s}) \mid \mathfrak{s} \in \mathfrak{S}_{iter}\}$ weight 1
BS-update:	$\mathfrak{S}_{upd} \leftarrow \{\mathfrak{s}_{bs}\}$ weight 1

The ant colony optimization metaheuristic

Successful ACO variant: Ant Colony System(ACS), [Gambardella, Dorigo, 1996]

Characteristic properties:

- Deterministic construction steps with probability q

$$\mathfrak{c} = \operatorname{argmax}_{\mathfrak{c}_i^j \in \mathfrak{N}(\mathfrak{s}^p)} [\tau_i^j]^{\alpha} \cdot [\eta(\mathfrak{c}_i^j)]^{\beta}$$

- Evaporation of pheromone during the construction of solution $\mathfrak{s}$:

$$\tau_i^j \leftarrow \gamma \tau_i^j + (1-\gamma)c \;\;, \forall \; \mathfrak{c}_i^j \in \mathfrak{s} \;\;,$$

  where $c > 0$ is the initial pheromone value, and $\gamma \in (0,1]$

- Use of the BS-update (evaporation only for used solution components)

The ant colony optimization metaheuristic

A general update rule:

$$\tau_i^j \leftarrow (1-\rho) \cdot \tau_i^j + \rho \cdot \sum_{\{\mathfrak{s} \in \mathfrak{S}_{upd} \mid \mathfrak{c}_i^j \in \mathfrak{s}\}} w_{\mathfrak{s}} \cdot F(\mathfrak{s}) \;\;,$$

where

- evaporation rate $\rho \in (0,1]$
- $\mathfrak{S}_{upd}$ is the set of solutions used for the update
- quality function $F : \mathfrak{S} \mapsto \mathbb{R}^+$. We use $F(\cdot) = \frac{1}{f(\cdot)}$
- $w_{\mathfrak{s}}$ is the weight of solution $\mathfrak{s}$

Question: Which solutions should be used for updating?

The ant colony optimization metaheuristic

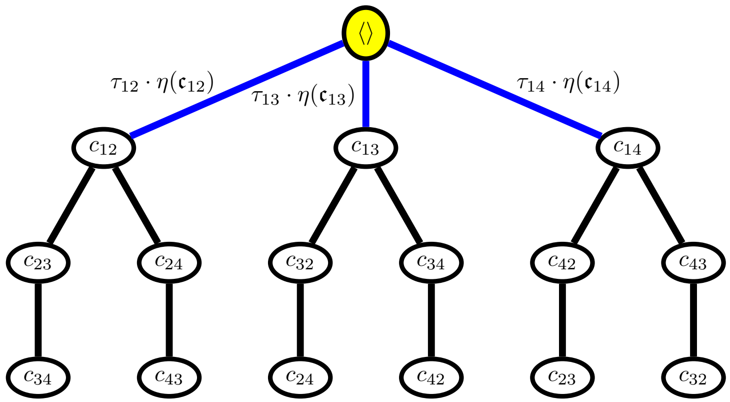
Successful ACO variant:
$\mathcal{MAX}$–$\mathcal{MIN}$ Ant System($\mathcal{MM}$AS), [Stützle, Hoos, 2000]

Characteristic properties:

- Use of a pheromone lower bound $\tau_{min} > 0$
- Application of restarts (by re-initializing the phermone values)
- Mix of IB-update and BS-update depending on a convergence measure

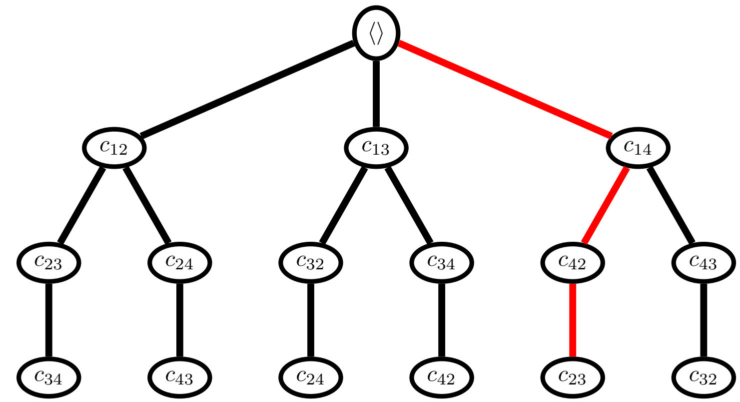
The ant colony optimization metaheuristic

ACO as a tree search algorithm: 1st construction step



The ant colony optimization metaheuristic

ACO as a tree search algorithm: 3rd construction step



The ant colony optimization metaheuristic

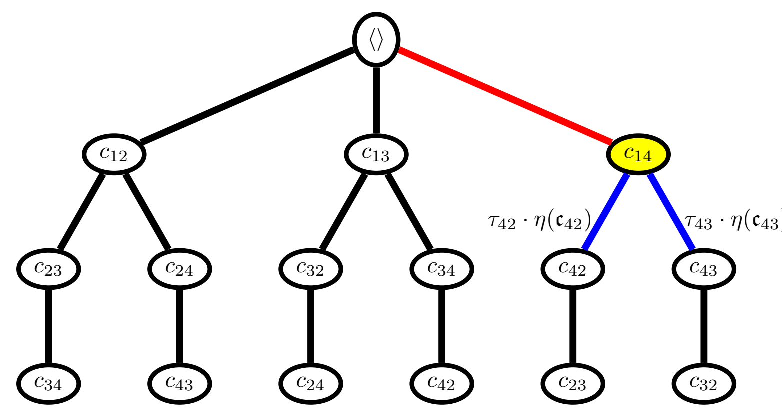
Hybridizations of ACO algorithms:

- Example 1: Hybridization with beam search [Blum, 2004]
- Example 2: Hybridization with constraint programming [Meyer, Ernst, 2004]
- Example 3: ACO and multi-level techniques [Korošec et al., 2004]

Important concept: ACO can be seen as a tree search method!

The ant colony optimization metaheuristic

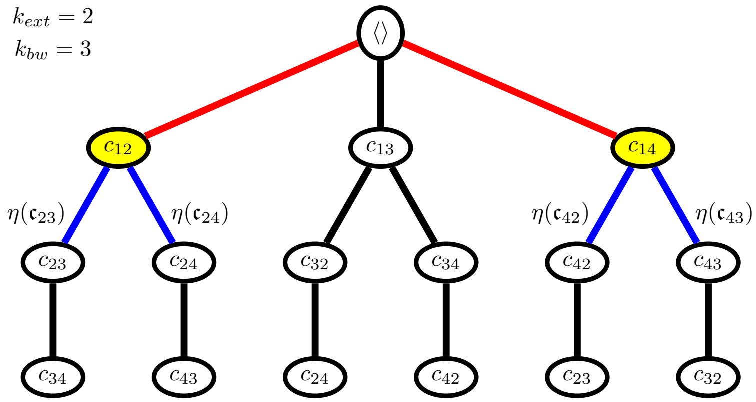
ACO as a tree search algorithm: 2nd construction step



The ant colony optimization metaheuristic

Beam search: 2nd construction step

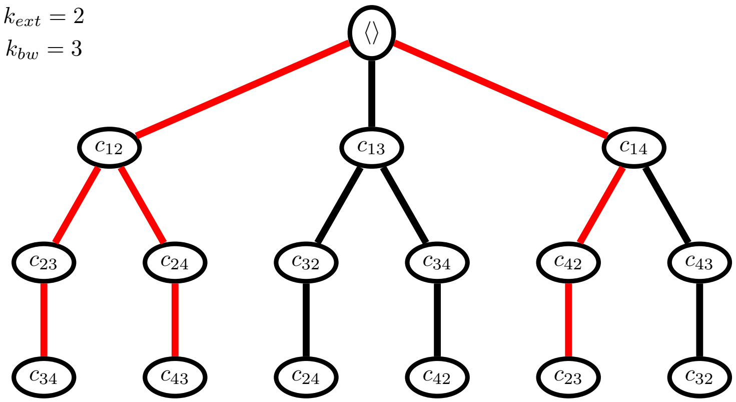
$k_{ext} = 2$
$k_{bw} = 3$



The ant colony optimization metaheuristic

Beam search: 3rd construction step

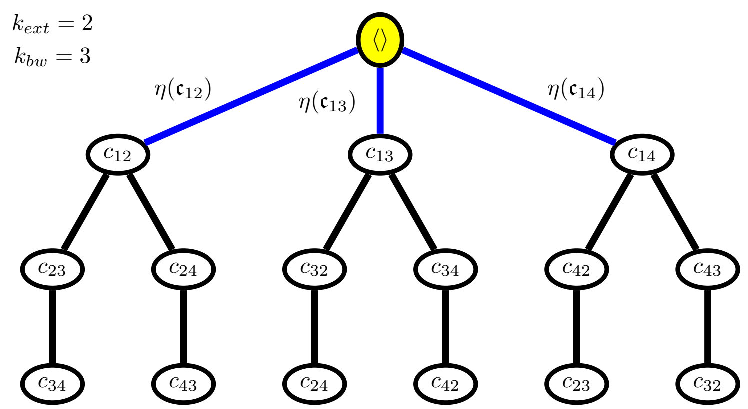
$k_{ext} = 2$
$k_{bw} = 3$



The ant colony optimization metaheuristic

Beam search: 1st construction step

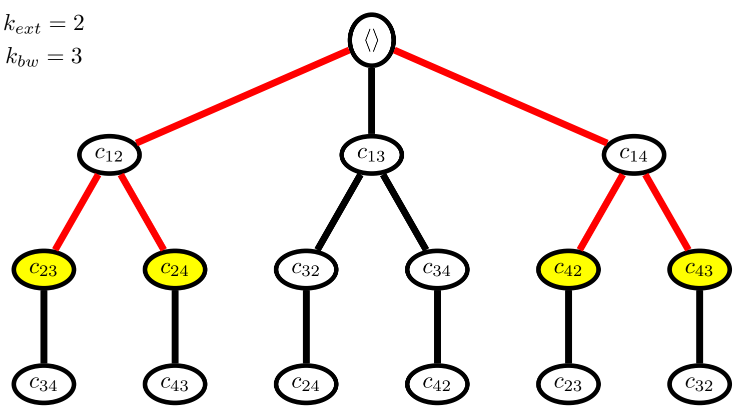
$k_{ext} = 2$
$k_{bw} = 3$



The ant colony optimization metaheuristic

Beam search: after 2nd construction step → use of lower bound

$k_{ext} = 2$
$k_{bw} = 3$



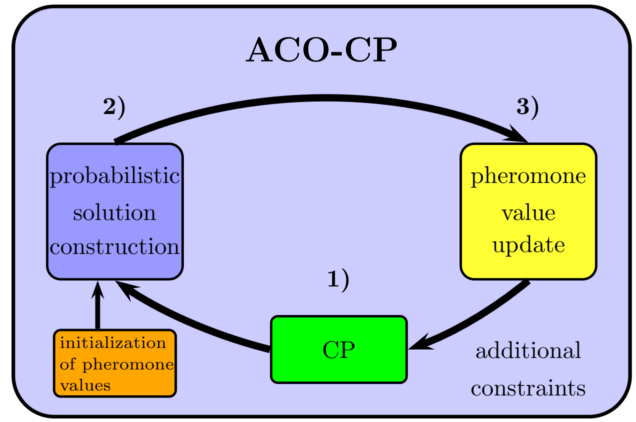
The ant colony optimization metaheuristic

Hybridizations of ACO algorithms:

- Example 1: Hybridization with beam search [Blum, 2004]
- Example 2: Hybridization with constraint programming [Meyer, Ernst, 2004]
- Example 3: ACO and multi-level techniques [Korošec et al., 2004]

The ant colony optimization metaheuristic

ACO-CP hybrid:



The ant colony optimization metaheuristic

Idea: Beam-ACO, in which each ant performs a probabilistic beam search

Advantages:

- Strong heuristic guidance by a lower bound
- Embedded in the adaptive framework of ACO

Result: Beam-ACO is state-of-the-art for open shop scheduling (OSS)

The ant colony optimization metaheuristic

Constraint programming (CP): Study of computational systems based on constraints

How does it work?

- Phase 1:
  - Express CO problem in terms of a model (variables+domains)
  - Define ("post") constraints among the variables
  - The constraint solver reduces the variable domains
- Phase 2: Labelling
  - Search through the remaining search tree
  - Possibly "post" additional constraints

The ant colony optimization metaheuristic

Hybridizations of ACO algorithms:

- Example 1: Hybridization with beam search [Blum, 2004]
- Example 2: Hybridization with constraint programming [Meyer, Ernst, 2004]
- Example 3: ACO and multi-level techniques [Korošec et al., 2004]

The ant colony optimization metaheuristic

Application fields of multi-level techniques:

- Originally: graph-based optimization problems
- In general:
  - When problem instances can be contracted while maintaining characteristics
  - When large-scale problem instances are considered

Multi-level ACO: Very good performance for mesh-partitioning.

The ant colony optimization metaheuristic

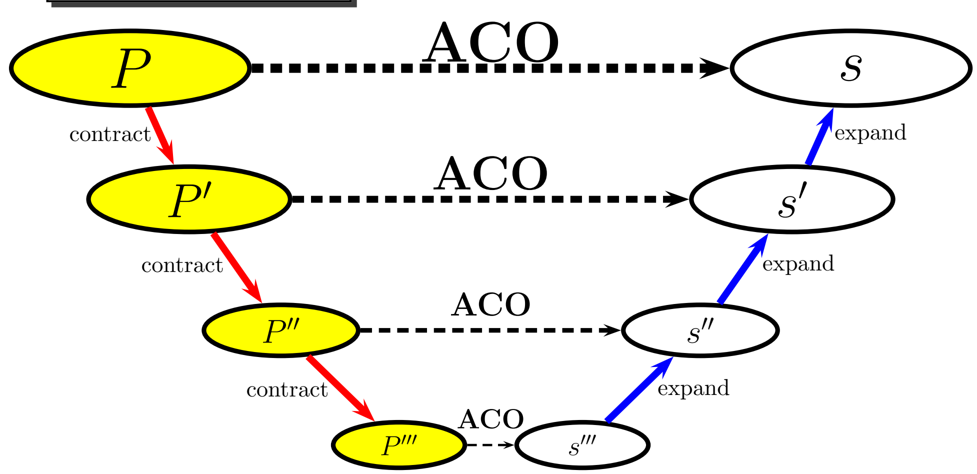
Advantages:

- Advantage of ACO:
  Good in finding high quality solutions for moderately constrained problems.
- Advantage of CP:
  Good in finding feasible solutions for highly constrained problems.

ACO-CP:
Promising for constrained problems with still a high number of feasible solutions.

The ant colony optimization metaheuristic

The multi-level framework:



Theoretical studies of ant colony optimization

Models of ACO algorithms:

- Application of the expected pheromone update
- Model notation: $M(< \text{problem} >, < \text{update\_rule} >, < \text{nr\_of\_ants} >)$
- $W_F(\mathcal{T})$: Expected iteration quality

Example of $W_F(\mathcal{T})$: $M(*, *, n_a = \infty)$

$$W_F(\mathcal{T}) = \sum_{\mathfrak{s} \in \mathfrak{S}} F(\mathfrak{s}) \cdot \mathbf{p}(\mathfrak{s} \mid \mathcal{T})$$

Theoretical studies of ant colony optimization

The problem to solve:

1. ACO point of view:

$$\text{Find } \mathfrak{s}^* \leftarrow \operatorname{argmax}_{\mathfrak{s} \in \mathfrak{S}} F(\mathfrak{s})$$

2. Model-based search point of view:

$$\text{Find } \tau^* \leftarrow \operatorname{argmax}_{\tau} W_F(\mathcal{T})$$

Ways to tackle problem 2):

1. (Stochastic) gradiend ascent (SGA)
2. The cross entropy method (CE)

Theoretical studies of ant colony optimization

- Connection to other algorithms: model-based search
- Negative search bias

Theoretical studies of ant colony optimization

Model-based search: [Zlochin et al., 2004]

Central component: A probabilistic model $M \in \mathcal{M}$
$\rightarrow$ model parameter values induce a probability distribution over $\mathcal{S}$

Iterate:

- Generate candidate solutions using the probability distribution
- Use the candidate solutions to modify the probabilistic model
  Aim: Bias future sampling to high quality solutions

Observation: Most algorithms use a fixed model structure $\rightarrow$ only model parameter values are changed

Theoretical studies of ant colony optimization

Stochastic gradient ascent (SGA):

$$\tau \leftarrow \tau + \alpha \sum_{\mathfrak{s} \in \mathfrak{S}_{upd}} F(\mathfrak{s}) \nabla \ln \mathbf{p}(\mathfrak{s} \mid \mathcal{T}) \ ,$$

where $\mathfrak{S}_{upd}$ is the sample of solutions

An SGA pheromone update for ACO algorithms: $\rightarrow$ calculate $\nabla \ln \mathbf{p}(\mathfrak{s} \mid \mathcal{T})$

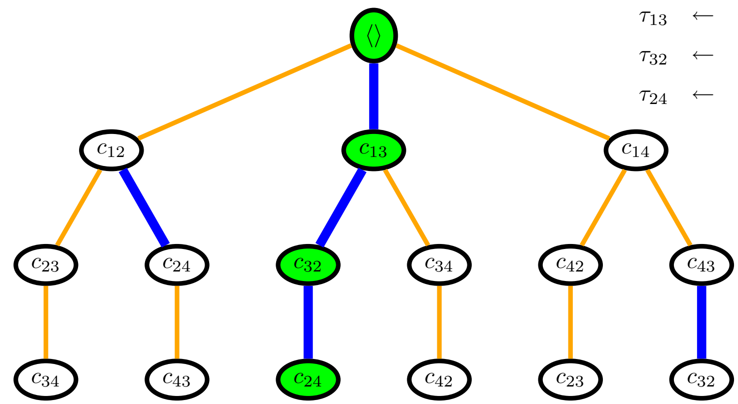
$$\nabla \ln \mathbf{p}(\mathfrak{s} \mid \mathcal{T}) = \nabla \ln \prod_{h=1}^{|\mathfrak{s}|-1} \mathbf{p}(\mathfrak{c}_{h+1} \mid \mathfrak{s}_h^p) = \sum_{h=1}^{|\mathfrak{s}|-1} \nabla \ln \mathbf{p}(\mathfrak{c}_{h+1} \mid \mathfrak{s}_h^p) \ ,$$

where $\mathfrak{s} = \langle \mathfrak{c}_1, \mathfrak{c}_2, \ldots, \mathfrak{c}_{|\mathfrak{s}|} \rangle$

Theoretical studies of ant colony optimization

ACO update:

$$\begin{aligned} \tau_{ij} &\leftarrow \rho \cdot \tau_{ij} \\ \tau_{13} &\leftarrow \tau_{13} + \rho F(\cdot) \\ \tau_{32} &\leftarrow \tau_{32} + \rho F(\cdot) \\ \tau_{24} &\leftarrow \tau_{24} + \rho F(\cdot) \end{aligned}$$



Theoretical studies of ant colony optimization

Gradient ascent:

$$\tau \leftarrow \tau + \alpha \nabla \, W_F(\mathcal{T})|_{\tau}$$

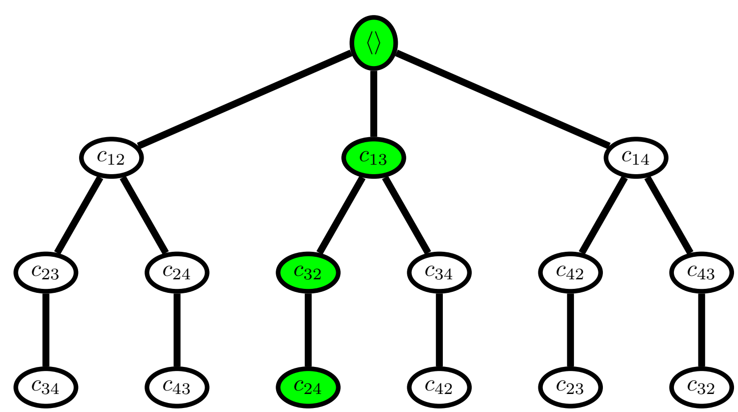
Calculation of the gradient:

$$\begin{aligned} \nabla W_F(\mathcal{T}) &= \nabla \sum_{\mathfrak{s} \in \mathfrak{S}} F(\mathfrak{s}) \mathbf{p}(\mathfrak{s} \mid \mathcal{T}) = \sum_{\mathfrak{s} \in \mathfrak{S}} F(\mathfrak{s}) \nabla \mathbf{p}(\mathfrak{s} \mid \mathcal{T}) \\ &= \sum_{\mathfrak{s} \in \mathfrak{S}} \mathbf{p}(\mathfrak{s} \mid \mathcal{T}) F(\mathfrak{s}) \frac{\nabla \mathbf{p}(\mathfrak{s} \mid \mathcal{T})}{\mathbf{p}(\mathfrak{s} \mid \mathcal{T})} \\ &= \sum_{\mathfrak{s} \in \mathfrak{S}} \mathbf{p}(\mathfrak{s} \mid \mathcal{T}) F(\mathfrak{s}) \nabla \ln \mathbf{p}(\mathfrak{s} \mid \mathcal{T}) \end{aligned}$$

Obervation: Calculation hardly possible, because $\mathfrak{S}$ is often too large

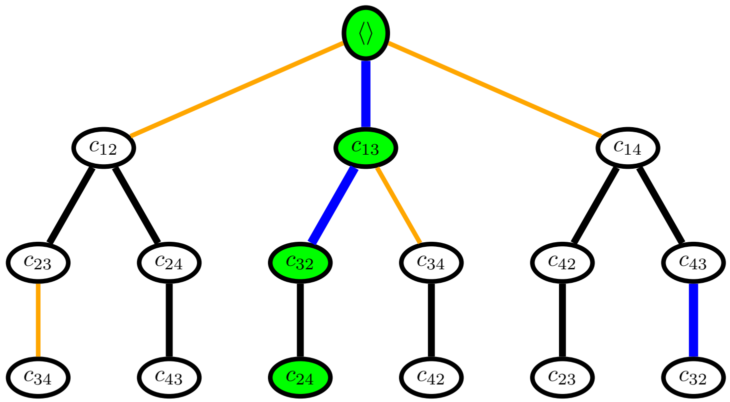
Theoretical studies of ant colony optimization

TSP example on 4 cities: Solution for update



Theoretical studies of ant colony optimization

SGA update:



Theoretical studies of ant colony optimization

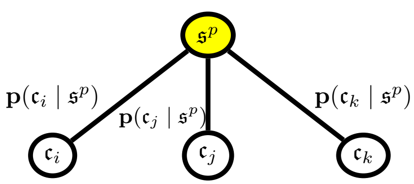
Search bias in ant colony optimization:

- Positive (and wanted) bias: Choice of (in comparison) good solutions for updating
- Negative bias:
  1. Modelling of the problem
  2. Solution construction process
  3. Pheromone update

How to detect negative bias? Decreasing algorithm performance over time

Theoretical studies of ant colony optimization

SGA update:



Assume: $\mathfrak{c}_j$ is chosen

Update:

- $\tau_i \leftarrow \tau_i + \rho(-\mathbf{p}(\mathfrak{c}_i \mid \mathfrak{s}^p))$
- $\tau_j \leftarrow \tau_j + \rho(1 - \mathbf{p}(\mathfrak{c}_j \mid \mathfrak{s}^p))$
- $\tau_k \leftarrow \tau_k + \rho(-\mathbf{p}(\mathfrak{c}_k \mid \mathfrak{s}^p))$

Theoretical studies of ant colony optimization

Differences between ACO and SGA-ACO:

- In SGA-ACO: No pheromone evaporation
- The pheromone value of "unused" solution components remains unchanged
- Pheromone values can be negative in SGA-ACO. Therefore:

$$\mathbf{p}(\mathfrak{c}_i^j \mid \mathfrak{s}^p) = \frac{e^{\left([\tau_i^j]^\alpha \cdot [\eta(\mathfrak{c}_i^j)]^\beta\right)}}{\sum\limits_{\mathfrak{c}_k^l \in \mathfrak{N}(\mathfrak{s}^p)} e^{\left([\tau_k^l]^\alpha \cdot [\eta(\mathfrak{c}_k^l)]^\beta\right)}} \ , \ \ \forall\ \mathfrak{c}_i^j \in \mathfrak{N}(\mathfrak{s}^p) \ ,$$

- Local search cannot be used as easily

Theoretical studies of ant colony optimization

Implicit assumptions in ACO:

Assumption 1:

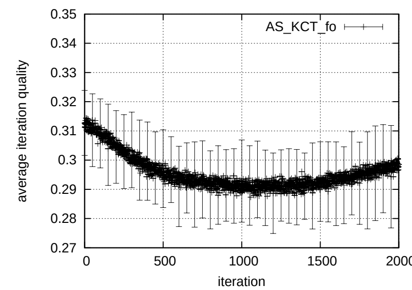
Good solutions are composed of good solution components.
(A solution component is regarded to be good, if the average quality of the solutions that contain it is high.)

Assumption 2:

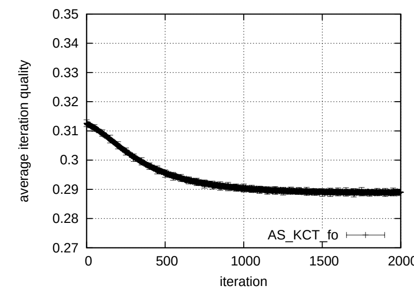
~~The pheromone update is such that good solution components on average are stronger reinforced than others.~~

Theoretical studies of ant colony optimization

Average iteration quality of Ant System $\rho = 0.01$



$n_a = 10$



$n_a = 1000$

Theoretical studies of ant colony optimization

Implicit assumptions in ACO:

Assumption 1:

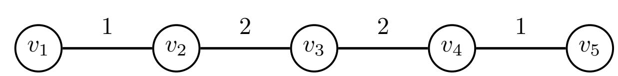
Good solutions are composed of good solution components.
(A solution component is regarded to be good, if the average quality of the solutions that contain it is high.)

Assumption 2:

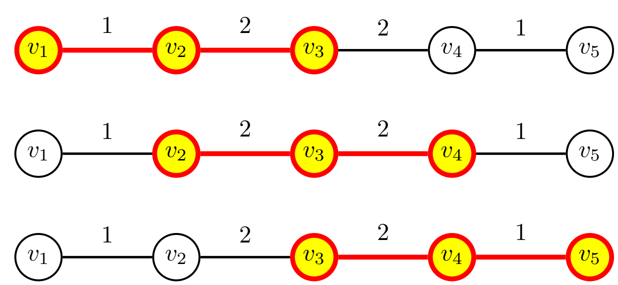
The pheromone update is such that good solution components on average are stronger reinforced than others.

Theoretical studies of ant colony optimization

Example: 2-cardinality tree problem



3 different solutions:



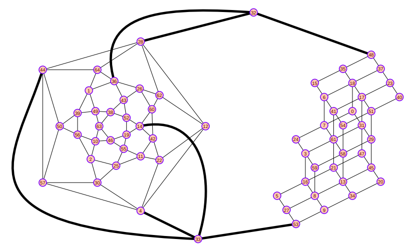
$\mathfrak{s}_1$: $f(\mathfrak{s}_1) = 3$

$\mathfrak{s}_2$: $f(\mathfrak{s}_2) = 4$

$\mathfrak{s}_3$: $f(\mathfrak{s}_3) = 3$

Theoretical studies of ant colony optimization

Instance statistics:



Theoretical studies of ant colony optimization

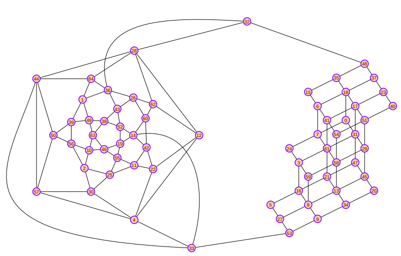
Conclusion: In case an ACO algorithm applied to a problem instance is **NOT** a competition-balanced system $\rightarrow$ possibility of negative search bias

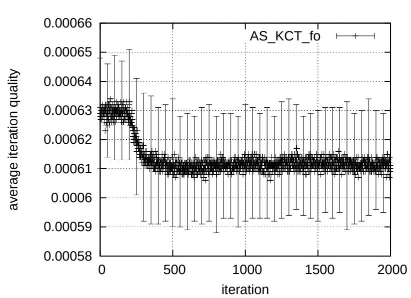
Existing theoretical results: The Ant System algorithm applied to unconstrained problems does not suffer from negative search bias

Open question Can it be shown that a competition-balanced system does not suffer from negative search bias?

Theoretical studies of ant colony optimization

Benchmark instances: Ant System applied to an Internet-like instance



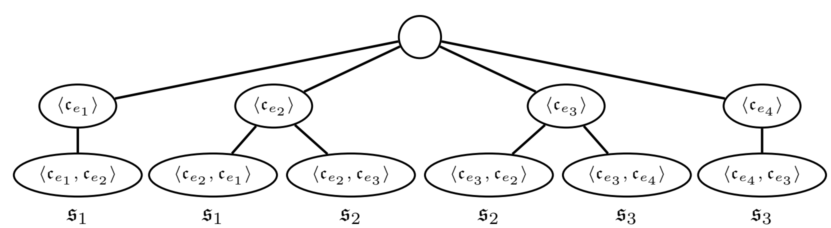


instance `gd96c` (65 nodes, 125 edges)

10 ants, $\rho = 0.1$, $k = 30$

Theoretical studies of ant colony optimization

Definition: Competition-balanced system (CBS) An ACO algorithm applied to $P \in \mathcal{P}$ is called a CBS, if the following holds: Given a feasible partial solution $\mathfrak{s}^p$ and the set of solution components $\mathfrak{N}(\mathfrak{s}^p)$ that can be added to extend the partial solution $\mathfrak{s}^p$, each solution component $\mathfrak{c} \in \mathfrak{N}(\mathfrak{s}^p)$ is a component of the same number of feasible solutions (in terms of sequences built by the algorithm) as each other solution component $\mathfrak{c}' \in \mathfrak{N}(\mathfrak{s}^p)$, $\mathfrak{c} \neq \mathfrak{c}'$.



Questions?