Theory of Swarm Intelligence

Dirk Sudholt

University of Sheffield, UK

Tutorial at GECCO 2012

Parts of the material used with kind permission by Heiko Röglin and Carsten Witt.

GECCO'12 Companion, July 7-11, 2012, Philadelphia, PA, USA. ACM 978-1-4503-1178-6/12/07.

Swarm Intelligence

Collective behavior of a "swarm" of agents.

Examples from Nature

- dome construction by termites
- communication of bees
- ant trails
- foraging behavior of fish schools and bird flocks
- swarm robotics

Plenty of inspiration for optimization.

ACO and PSO

Ant colony optimization (ACO)

- inspired by foraging behavior of ants
- artificial ants construct solutions using pheromones
- pheromones indicate attractiveness of solution component

Particle swarm optimization (PSO)

- mimics search of bird flocks and fish schools
- particles "fly" through search space
- each particle is attracted by own best position and best position of neighbors

Theory

What "theory" can mean

- convergence analysis
- analysis of simplified models of algorithms
- empirical studies on test functions
- runtime analysis / computational complexity analysis
- ...

Example Question

How long does it take on average until algorithm A finds a target solution on problem P?

Notion of time: number of iterations, number of function evaluations

Content

What this tutorial is about

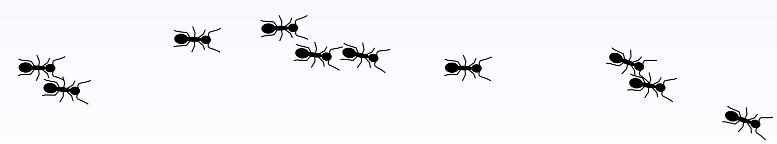
- runtime analysis
- simple variants of swarm intelligence algorithms
- insight into their working principles
- impact of parameters and design choices on performance
- what distinguishes ACO/PSO from evolutionary algorithms?
- performance guarantees for combinatorial optimization
- methods and proof ideas

What this tutorial is not about

- convergence results
- analysis of models of algorithms
- no intend to be exhaustive

Overview

Ant Colony Optimization (ACO)



Main idea: artificial ants communicate via pheromones.

Scheme of ACO

Repeat:

- construct ant solutions guided by pheromones
- update pheromones by reinforcing good solutions

Pseudo-Boolean Optimization

Goal: maximize $f \colon \{0,1\}^n \to \mathbb{R}$.

Illustrative test functions

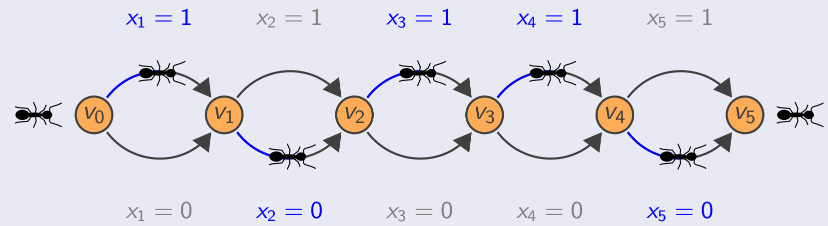
$$\textsc{OneMax}(x) = \sum_{i=1}^{n} x_i$$

$$\textsc{LeadingOnes}(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$$

$$\textsc{Needle}(x) = \prod_{i=1}^{n} x_i$$

ACO in Pseudo-Boolean Optimization

Solution Construction



Probability of choosing an edge equals pheromone on the edge.

Initial pheromones: $\tau(x_i = 0) = \tau(x_i = 1) = 1/2$.

Note: no linkage between bits. No heuristic information used.

Pheromones $\tau(x_i = 1)$ suffice to describe all pheromones.

ACO in Pseudo-Boolean Optimization (2)

Pheromone update: reinforce some good solution x.

Strength of update determined by evaporation factor $0 \leq \rho \leq 1$:

$$\tau'(x_i = 1) = \begin{cases} (1-\rho) \cdot \tau(x_i = 1) & \text{if } x_i = 0 \\ (1-\rho) \cdot \tau(x_i = 1) + \rho & \text{if } x_i = 1 \end{cases}$$

Pheromone borders as in MAX-MIN Ant System (Stützle and Hoos, 2000):

$$\tau_{\min} \leq \tau' \leq 1 - \tau_{\min}$$

Default choice: $\tau_{\min} := 1/n$ (cf. standard mutation in EAs).

One Ant?

Most ACO algorithms analyzed: one ant per iteration.

One ant at a time, many ants over time.

Steady-state GA

- Probabilistic model: Population
- New solutions: selection + variation
- Environmental selection

Ant Colony Optimization

- Probabilistic model: Pheromones
- New solutions: construction graph
- Selection for reinforcement

Evolutionary Algorithms vs. ACO

MMAS* (Gutjahr and Sebastiani, 2008)

Start with uniform random solution x^* and repeat:

- Construct x.
- Replace x^* by x if $f(x) > f(x^*)$.
- Update pheromones w. r. t. x^* (best-so-far update).

Note: best-so-far solution x^* is constantly reinforced.

(1+1) EA

Start with uniform random solution x^* and repeat:

- Create x by flipping each bit in x^* independently with probability $1/n$.
- Replace x^* by x if $f(x) \geq f(x^*)$.

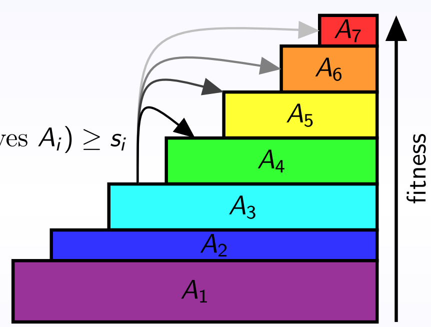
(1+1) EA: Probability of setting bit to 1 is in $\{1/n, 1-1/n\}$.

MMAS*: Probability of setting bit to 1 is in $[1/n, 1-1/n]$ (unless $\rho \approx 1$).

Overview

Fitness-level Method for the (1+1) EA

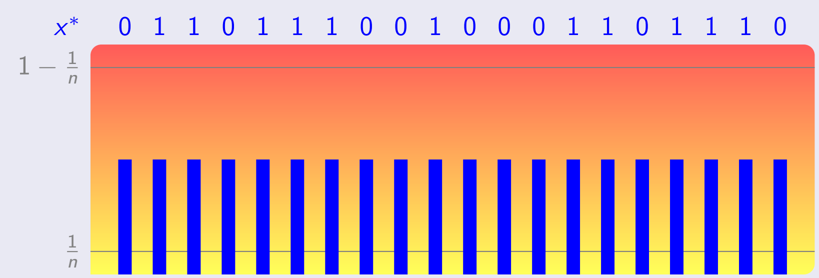
$\Pr((1+1)\text{ EA leaves } A_i) \geq s_i$



Expected optimization time of (1+1) EA at most $\sum_{i=1}^{m-1} \frac{1}{s_i}$.

MMAS*

Pheromones on 1-edges



After $(\ln n)/\rho$ reinforcements of x^* MMAS* temporarily behaves like (1+1) EA.

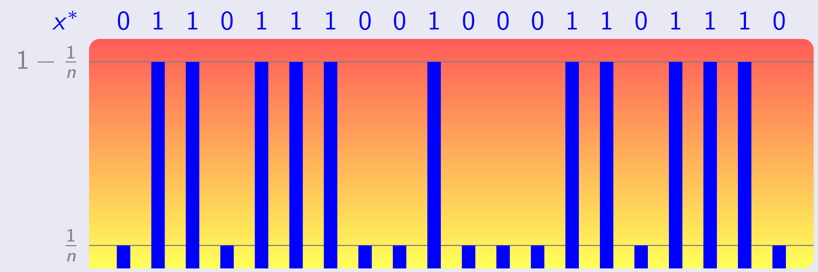
Fitness-Level Method with A_i = search points with i-th fitness value

(1+1) EA: $\leq \sum_{i=1}^{m-1} \frac{1}{s_i}$ MMAS*: $\leq \sum_{i=1}^{m-1} \frac{1}{s_i} + m \cdot \frac{\ln n}{\rho}$

Upper bounds: time for finding improvements + time for pheromone adaptation.

MMAS*

Pheromones on 1-edges



After $(\ln n)/\rho$ reinforcements of x^* MMAS* temporarily behaves like (1+1) EA.

Fitness-Level Method with A_i = search points with i-th fitness value

(1+1) EA: $\leq \sum_{i=1}^{m-1} \frac{1}{s_i}$ MMAS*: $\leq \sum_{i=1}^{m-1} \frac{1}{s_i} + m \cdot \frac{\ln n}{\rho}$

Upper bounds: time for finding improvements + time for pheromone adaptation.

Bounds with Fitness Levels

OneMax:

$$s_i \geq (n-i) \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{n-i}{en}$$

Theorem

$$\text{(1+1) EA:} \quad en \sum_{i=0}^{n-1} \frac{1}{n-i} = O(n \log n)$$

$$\text{MMAS*:} \quad en \sum_{i=0}^{n-1} \frac{1}{n-i} + n \cdot \frac{\ln n}{\rho} = O((n \log n)/\rho)$$

Bounds with Fitness Levels (2)

LeadingOnes

$$s_i \geq \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{en}$$

Theorem

$$\text{(1+1) EA:} \quad en^2 \qquad \text{MMAS*:} \quad en^2 + n \cdot \frac{\ln n}{\rho} = O(n^2 + (n \log n)/\rho)$$

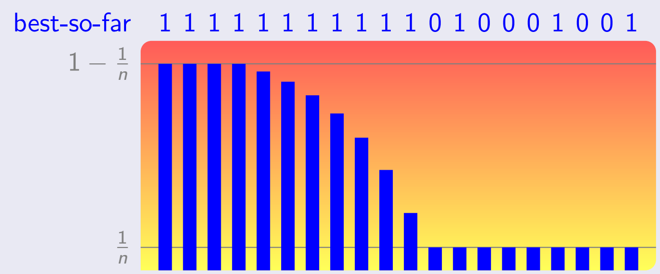
Unimodal functions with d function values:

Theorem

$$\text{(1+1) EA:} \quad end \qquad \text{MMAS*:} \quad end + d \cdot \frac{\ln n}{\rho} = O(nd + (d \log n)/\rho)$$

Layering of Pheromones for LeadingOnes

(Lower bounds on) pheromones on LeadingOnes



Improved results for

- LeadingOnes (Neumann, Sudholt, and Witt, 2009)
- shortest paths (Sudholt and Thyssen, 2012)
- Binary PSO on OneMax (Sudholt and Witt, 2010)

Overview

Strict Selection

Most ACO algorithms replace x^* only if $f(x) > f(x^*)$.

Drawback

Cannot explore plateaus.

Theorem (Neumann, Sudholt, Witt, 2009)

If $\rho \geq 1/poly(n)$ the expected time of MMAS* *on* NEEDLE *is* $\Omega(2^{-n} \cdot n^n) = \Omega((n/2)^n)$.

Define variant MMAS of MMAS* replacing x^* if $f(x) \geq f(x^*)$.
Pheromones on each bit perform a random walk.

Theorem (Neumann, Sudholt, Witt, 2009 and Sudholt, 2011)

The expected time of MMAS *on* NEEDLE *is* $O(n^2/\rho^2 \cdot \log n \cdot 2^n)$.

Mixing time estimates (Sudholt, 2011)

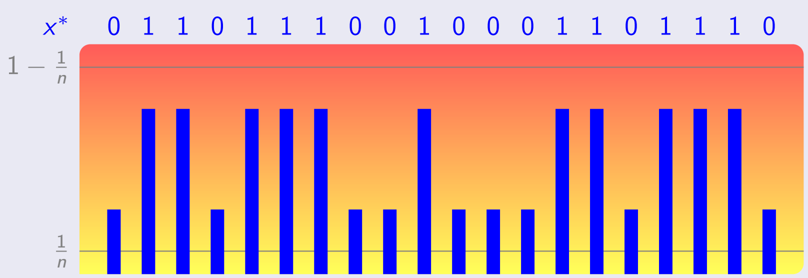
MMAS "forgets" initial pheromones on bits that have been irrelevant for the last $\Omega(n^2/\rho^2)$ steps.

MMAS and Fitness Levels

How does MMAS cope with plateaus on fitness levels?

Switching between equally fit solutions can prevent freezing.

Pheromones on 1-edges



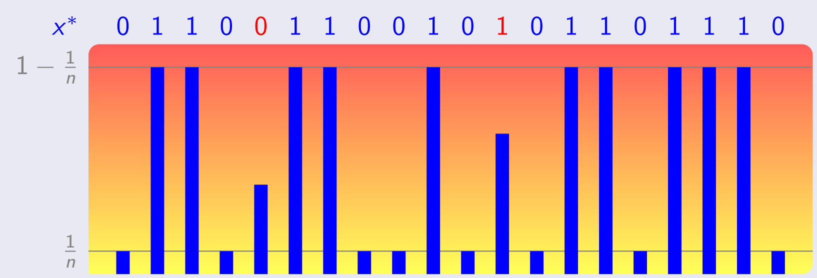
Fitness-level method breaks down!

MMAS and Fitness Levels

How does MMAS cope with plateaus on fitness levels?

Switching between equally fit solutions can prevent freezing.

Pheromones on 1-edges



Fitness-level method breaks down!

Is this Behavior Detrimental?

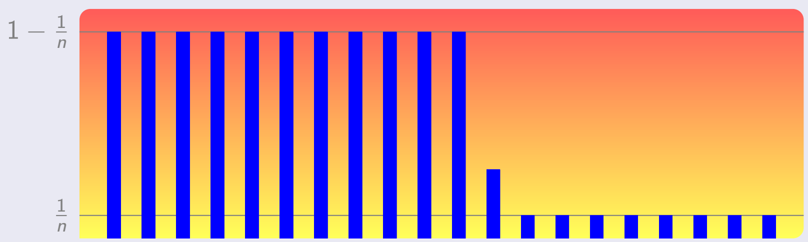
Probably not.

Theorem (Kötzing, Neumann, Sudholt, and Wagner, 2011)

$O(n \log n + n/\rho)$ on OneMax *for both* MMAS* *and* MMAS.

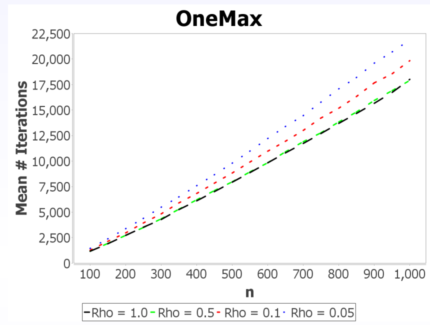
Assuming the sum of pheromones is fixed. Worst possible pheromone distribution for finding improvements on OneMax (Gleser, 1975):

Pheromones on 1-edges

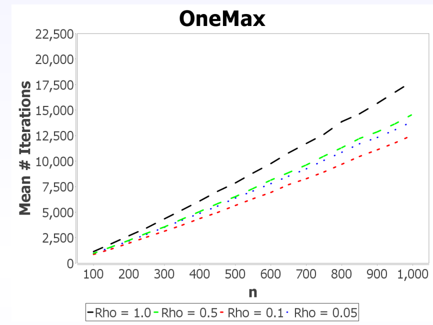


Worst case: all pheromones (but one) at borders.

Experiments (Kötzing et al., 2011)



MMAS*



MMAS

- MMAS better than MMAS*
- MMAS with $\rho < 1$ better than (1+1) EA (=MMAS at $\rho = 1$)!
- does not hold for MMAS*

Open Problem

Prove that MMAS with proper ρ is faster than MMAS* and (1+1) EA.

Overview

Iteration-Best Update

λ-MMAS$_{\text{ib}}$

Repeat:
- construct λ ant solutions
- update pheromones by reinforcing the best of these solutions

Advantages:
- can escape from local optima
- inherently parallel
- simpler ants

Iteration-Best vs. Comma Strategies

Jägersküpper and Storch, 2007

(1,λ) EA: $\lambda \geq c \log n$ necessary, even for OneMax.

If $\lambda \leq c' \log n$ then (1,λ) EA needs exponential time.

Reason: (1,λ) EA moves away from optimum if close and λ too small.

Behavior too chaotic to allow for hill climbing!

Iteration-Best on OneMax

Slow pheromone adaptation effectively eliminates chaotic behavior.

Theorem (Neumann, Sudholt, and Witt, 2010)

If $\rho \leq 1/(cn^{1/2} \log n))$ for a sufficiently large constant $c > 0$ and $\rho \geq 1/poly(n)$ then 2-MMAS$_{ib}$ optimizes OneMax in expected time $O(\sqrt{n}/\rho)$.
For $\rho = 1/(cn^{1/2} \log n)$ the time bound is $O(n \log n)$.

Two ants are enough!

Proof idea: as long as all pheromones are at least 1/3, the sum of pheromones grows steadily.

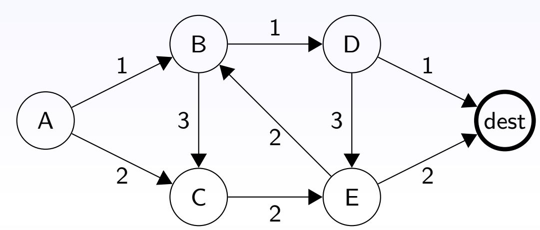
Large ρ or small λ: pheromones come crashing down to $1/n$.

Theorem

Choosing $\lambda/\rho \leq (\ln n)/244$, the optimization time of λ-MMAS$_{ib}$ on every function with a unique optimum is $2^{\Omega(n^{\varepsilon})}$ for some constant $\varepsilon > 0$ w. o. p.

Overview

ACO System for Single-Destination Shortest Path Problem

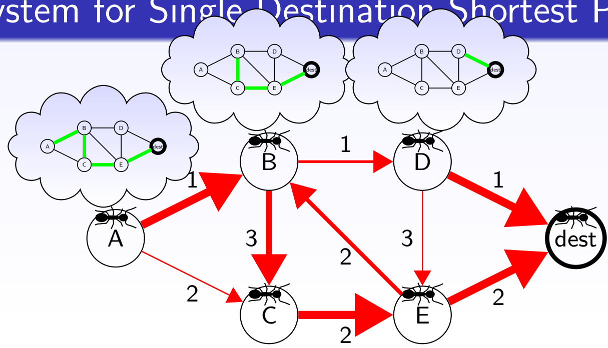


MMAS$_{SDSP}$

For each vertex u the ant

- memorizes and keeps track of its best-so-far path
- constructs a simple path from u to the destination
- updates pheromones on edges $(u, \cdot)$ (local pheromone update)

ACO System for Single Destination Shortest Path Problem

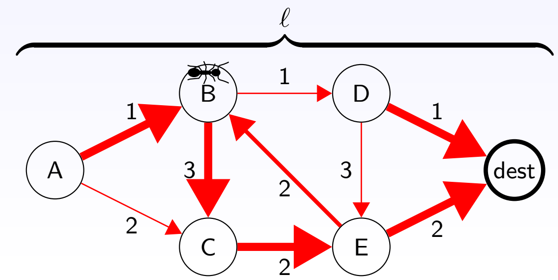


MMAS$_{\text{SDSP}}$

For each vertex u the ant

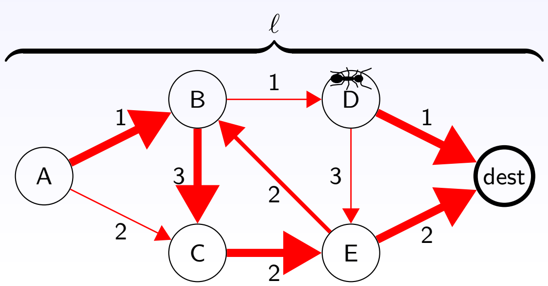
- memorizes and keeps track of its best-so-far path
- constructs a simple path from u to the destination
- updates pheromones on edges $(u, \cdot)$ (local pheromone update)

Shortest Paths Propagate Through the Graph



Let $\tau_{\min} := 1/(\Delta\ell)$. Consider vertex u such that all ants on its shortest paths have found shortest paths and adapted their pheromones.

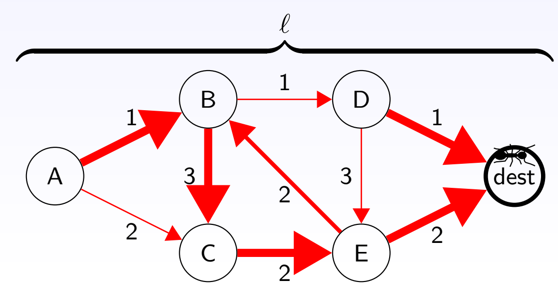
Shortest Paths Propagate Through the Graph



Let $\tau_{\min} := 1/(\Delta\ell)$. Consider vertex u such that all ants on its shortest paths have found shortest paths and adapted their pheromones.

- probability of ant at u choosing the first edge correctly $\geq \tau(e)/2 \geq \tau_{\min}/2$

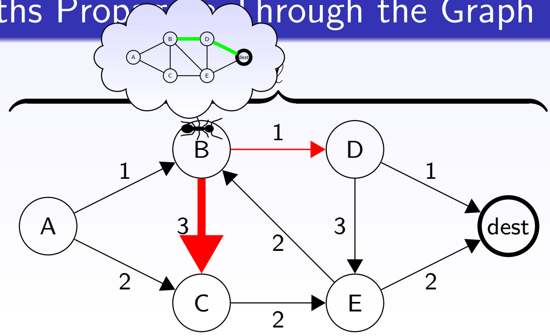
Shortest Paths Propagate Through the Graph



Let $\tau_{\min} := 1/(\Delta\ell)$. Consider vertex u such that all ants on its shortest paths have found shortest paths and adapted their pheromones.

- probability of ant at u choosing the first edge correctly $\geq \tau(e)/2 \geq \tau_{\min}/2$
- probability of following adapted pheromones: $(1 - 1/\ell)^{\ell-1} \geq 1/e$.

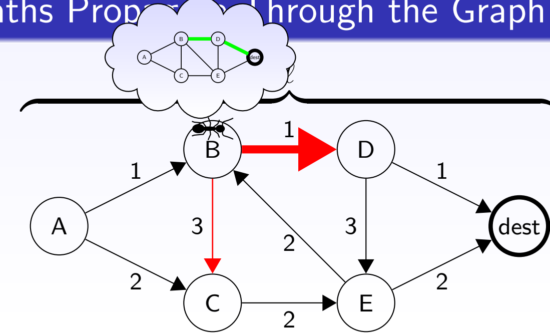
Shortest Paths Propagate Through the Graph



Let $\tau_{\min} := 1/(\Delta\ell)$. Consider vertex u such that all ants on its shortest paths have found shortest paths and adapted their pheromones.

- probability of ant at u choosing the first edge correctly $\geq \tau(e)/2 \geq \tau_{\min}/2$
- probability of following adapted pheromones: $(1-1/\ell)^{\ell-1} \geq 1/e$.

Shortest Paths Propagate Through the Graph

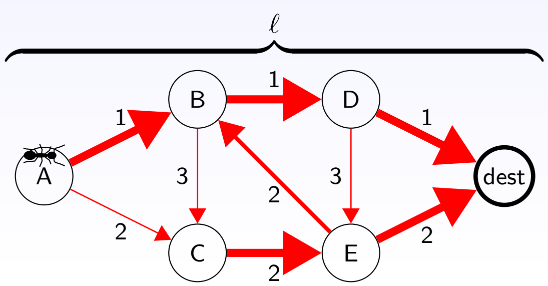


Let $\tau_{\min} := 1/(\Delta\ell)$. Consider vertex u such that all ants on its shortest paths have found shortest paths and adapted their pheromones.

- probability of ant at u choosing the first edge correctly $\geq \tau(e)/2 \geq \tau_{\min}/2$
- probability of following adapted pheromones: $(1-1/\ell)^{\ell-1} \geq 1/e$.

Expected time until ant at u has done the same $\leq 2e/\tau_{\min} + \ln(1/\tau_{\min})/\rho$.

Shortest Paths Propagate Through the Graph



Let $\tau_{\min} := 1/(\Delta\ell)$. Consider vertex u such that all ants on its shortest paths have found shortest paths and adapted their pheromones.

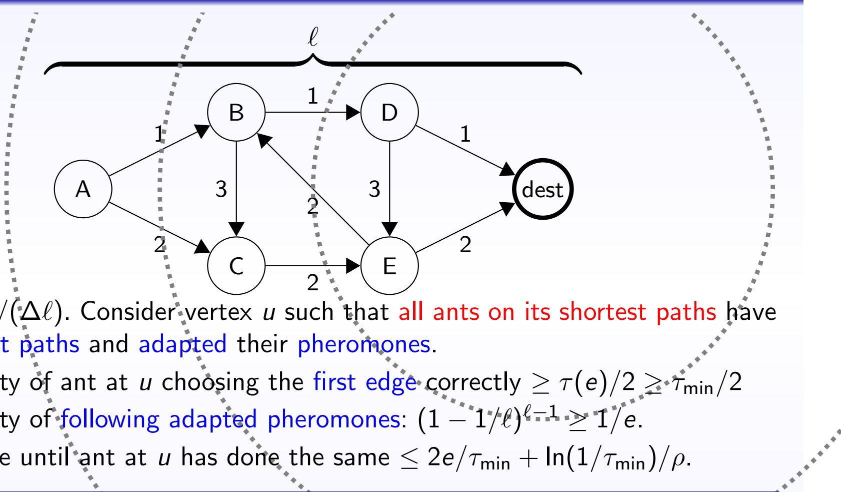
- probability of ant at u choosing the first edge correctly $\geq \tau(e)/2 \geq \tau_{\min}/2$
- probability of following adapted pheromones: $(1-1/\ell)^{\ell-1} \geq 1/e$.

Expected time until ant at u has done the same $\leq 2e/\tau_{\min} + \ln(1/\tau_{\min})/\rho$.

Upper bounds for MMAS$_{\text{SDSP}}$ (Sudholt and Thyssen, 2012)

- Consider all vertices sequentially: $O(n\Delta\ell + n\ln(\Delta\ell)/\rho)$.

Shortest Paths Propagate Through the Graph



Let $\tau_{\min} := 1/(\Delta\ell)$. Consider vertex u such that all ants on its shortest paths have found shortest paths and adapted their pheromones.

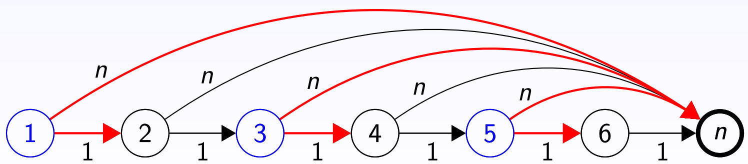
- probability of ant at u choosing the first edge correctly $\geq \tau(e)/2 \geq \tau_{\min}/2$
- probability of following adapted pheromones: $(1-1/\ell)^{\ell-1} \geq 1/e$.

Expected time until ant at u has done the same $\leq 2e/\tau_{\min} + \ln(1/\tau_{\min})/\rho$.

Upper bounds for MMAS$_{\text{SDSP}}$ (Sudholt and Thyssen, 2012)

- Consider all vertices sequentially: $O(n\Delta\ell + n\ln(\Delta\ell)/\rho)$.
- Slice graph into "layers" and use pheromone layering: $O(\Delta\ell^2 + \ell/\rho)$.

A Worst-Case Graph



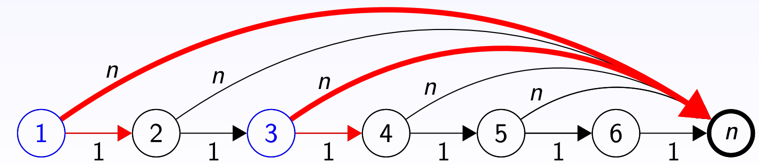
Expected time $O(\Delta\ell^2 + \ell/\rho)$ and $\Omega\left(\Delta\ell^2 + \frac{\ell}{\rho\log(1/\rho)}\right)$

- #wrong vertices decreases on average by $O(\rho\log(1/\rho))$.
- expected time for decrease of $\Omega(\ell) \Rightarrow \Omega\left(\frac{\ell}{\rho\log(1/\rho)}\right)$.

After pheromone adaptation still $\Omega(\ell)$ wrong vertices left

- #wrong vertices decreases on average by $O(\tau_{\min})$
- expected time for decrease of $\Omega(\ell) \Rightarrow \Omega\left(\frac{\ell}{\tau_{\min}}\right) = \Omega(\Delta\ell^2)$.

A Worst-Case Graph



Expected time $O(\Delta\ell^2 + \ell/\rho)$ and $\Omega\left(\Delta\ell^2 + \frac{\ell}{\rho\log(1/\rho)}\right)$

- #wrong vertices decreases on average by $O(\rho\log(1/\rho))$.
- expected time for decrease of $\Omega(\ell) \Rightarrow \Omega\left(\frac{\ell}{\rho\log(1/\rho)}\right)$.

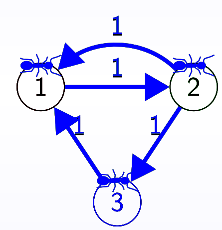
After pheromone adaptation still $\Omega(\ell)$ wrong vertices left

- #wrong vertices decreases on average by $O(\tau_{\min})$
- expected time for decrease of $\Omega(\ell) \Rightarrow \Omega\left(\frac{\ell}{\tau_{\min}}\right) = \Omega(\Delta\ell^2)$.

Overview

All-Pairs Shortest Path Problem

Use distinct pheromone function $\tau_v \colon E \to \mathbb{R}_0^+$ for each destination v:



A Simple Interaction Mechanism

Path construction with interaction

For each ant traveling from u to v
- with prob. $1/2$
  - use τ_v to travel from u to v
- with prob. $1/2$
  - choose an intermediate destination $w \in V$ uniformly at random
  - uses τ_w to travel from u to w
  - uses τ_v to travel from w to v

Speed-up by Interaction

Theorem

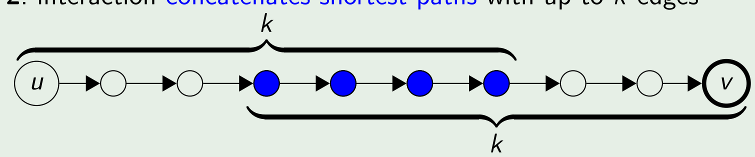
If $\tau_{\min} = 1/(\Delta\ell)$ and $\rho \leq 1/(23\Delta \log n)$ the number of iterations using interaction w. h. p. is $O(n \log n + \log(\ell) \log(\Delta\ell)/\rho)$.

Possible improvement: $O(n^3) \to O(n \log^3 n)$

Proof Sketch

Phase 1: find all shortest paths with one edge
slow evaporation $\longrightarrow$ near-uniform search

Phase 2: interaction concatenates shortest paths with up to k edges



$\longrightarrow$ find shortest paths with up to $3/2 \cdot k$ edges.

Note: slow adaptation helps!

Overview

Stochastic Shortest Paths

Directed acyclic graph $G = (V, E, w)$ with non-negative weights

For a path $p = (e_1, \ldots, e_\ell)$

$w(p) := \sum_{i=1}^{\ell} w(e_i)$ is the real length of p.

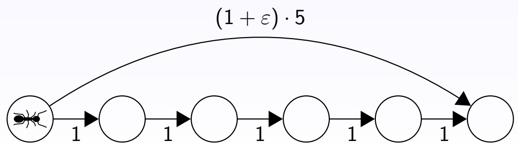
$\tilde{w}(p) := \sum_{i=1}^{\ell} (1 + \eta(e_i)) \cdot w(e_i)$ is the noisy length of p.

Goal

Find or approximate real shortest paths despite noise.

Ants Become Risk-Seeking

Every edge has independent noise $\sim \Gamma(k, \theta)$.



Algorithm: $\text{MMAS}_{\text{SDSP}}$, no re-evaluation of best-so-far paths.

Ant tends to store path with high variance as best-so-far path.

Theorem (Sudholt and Thyssen, 2012)

There is a graph where with probability $1 - \exp(-\Omega(\sqrt{n}/\log n))$ $\text{MMAS}_{\text{SDSP}}$ does not find a $(1 + k\theta/3)$-approximation for all vertices within e^{cn} iterations.

Re-evaluate best-so-far paths: Doerr, Hota, and Kötzing (this GECCO).

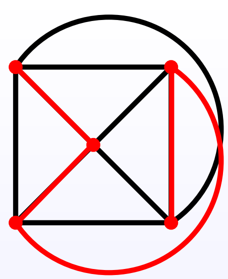
Overview

Broder's Algorithm

Problem: Minimum Spanning Trees

Consider the input graph itself as construction graph.

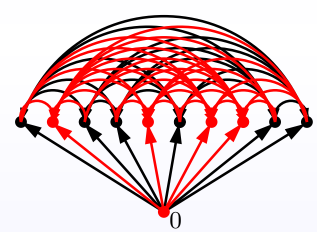
Spanning tree can be chosen uniformly at random using random walk algorithms (e. g. Broder, 1989).



Reward chosen edges ⇒ next solution will be similar to constructed one
But: local improvements are possible

Component-based Construction Graph

- Vertices correspond to edges of the input graph
- Construction graph $C(G) = (N, A)$ satisfies $N = \{0, \ldots, m\}$ (start vertex 0) and $A = \{(i,j) \mid 0 \le i \le m, 1 \le j \le m, i \ne j\}$.



For a given path $v_1, \ldots, v_k$ select the next edge from its neighborhood $N(v_1, \ldots, v_k) := (E \setminus \{v_1, \ldots, v_k\}) \setminus \{e \in E \mid (V, \{v_1, \ldots, v_k, e\})$ contains a cycle$\}$ (problem-specific aspect of ACO). Reward: all edges, that point to visited vertices (neglect order of chosen edges)

Algorithm

1-ANT: (following Neumann/Witt, 2010)

- two pheromone values
- value h: if edge has been rewarded
- value ℓ: otherwise
- heuristic information η, $\eta(e) = \frac{1}{w(e)}$ (used before for TSP)
- Let v_k the current vertex and N_{v_k} be its neighborhood.
- Prob(to choose neighbor y of v_k) $= \frac{[\tau_{(v_k,y)}]^\alpha \cdot [\eta_{(v_k,y)}]^\beta}{\sum_{y \in N(v_k)} [\tau_{(v_k,y)}]^\alpha \cdot [\eta_{(v_k,y)}]^\beta}$ with $\alpha, \beta \geq 0$.
- Consider special cases where either $\beta = 0$ or $\alpha = 0$.

Results for Pheromone Updates

Case $\alpha = 1$, $\beta = 0$: proportional influence of pheromone values

Theorem (Broder-based construction graph)

Choosing $h/\ell = n^3$, the expected time until the 1-ANT *with the Broder-based construction graph has found an MST is $O(n^6(\log n + \log w_{max}))$.*

Theorem (Component-based construction graph)

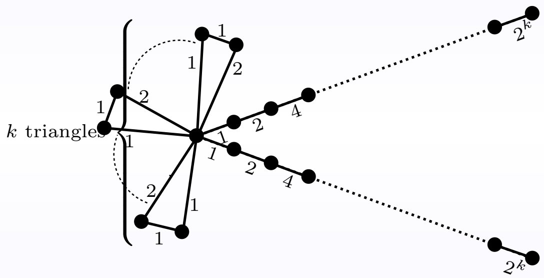
Choosing $h/\ell = (m - n + 1) \log n$, the expected time until the 1-ANT *with the component-based construction graph has found an MST is $O(mn(\log n + \log w_{max}))$.*

Better than (1+1) EA!

Broder Construction Graph: Heuristic Information

Example graph G^* with $n = 4k + 1$ vertices.

- k triangles of weight profile $(1, 1, 2)$
- two paths of length k with exponentially increasing weights.



Theorem (Broder-based construction graph)

Let $\alpha = 0$ and β be arbitrary, then the probability that the 1-ANT *using the Broder construction procedure does not find an MST in polynomial time is $1 - 2^{-\Omega(n)}$.*

Component-based Construction Graph/Heuristic Information

Theorem (Component-based construction graph)

Choosing $\alpha = 0$ and $\beta \geq 6w_{\max} \log n$, the expected time of the 1-ANT *with the component-based construction graph to find an MST is constant.*

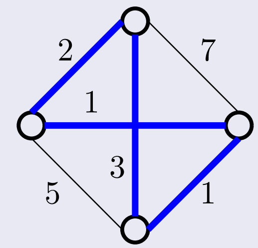
Proof Idea

- Choose edges as Kruskal's algorithm.
- Calculation shows: probability of choosing a lightest edge is at least $1 - 1/n$.
- $n - 1$ steps $\Longrightarrow$ probability for an MST is $\Omega(1)$.

Overview

Traveling Salesman Problem

Traveling Salesman Problem (TSP)



- Input: weighted complete graph $G = (V, E, w)$ with $w : E \to \mathbb{R}$.
- Goal: Find Hamiltonian cycle of minimum weight.

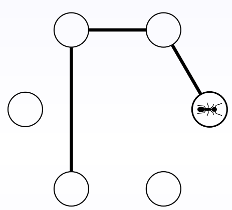
MMAS for TSP (Kötzing, Neumann, Röglin, Witt 2010)

Best-so-far pheromone update with $\tau_{\text{min}} := 1/n^2$ and $\tau_{\text{max}} := 1 - 1/n$.

Initialization: same pheromone on all edges.

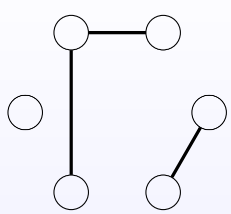
"Ordered" tour construction

Append a feasible edge chosen with probability proportional to pheromones.



"Arbitrary" tour construction

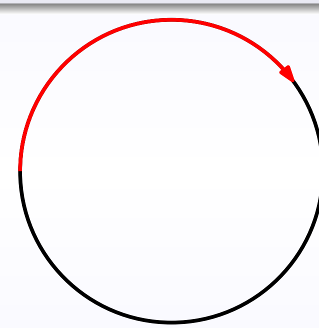
Add an edge chosen with probability proportional to pheromones as long as no cycle is closed or a vertex get degree at least 3.



Locality

Lemma

MMAS* with saturated pheromones exchanges $\Omega(\log(n))$ edges in expectation.



Length of unseen part roughly halves each time.

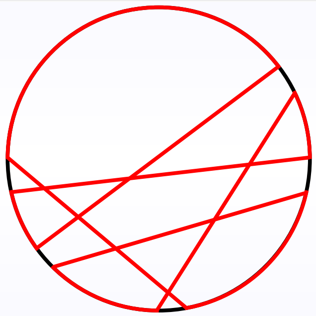
Lemma

For any constant k: MMAS^*_{Arb} with saturated pheromones creates exactly k new edges with probability $\Theta(1)$.

Locality

Lemma

MMAS* with saturated pheromones exchanges $\Omega(\log(n))$ edges in expectation.



Length of unseen part roughly halves each time.

Lemma

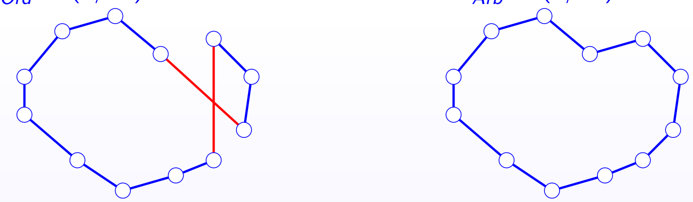
For any constant k: MMAS$^*_{Arb}$ with saturated pheromones creates exactly k new edges with probability $\Theta(1)$.

ACO Simulating 2-OPT

Zhou (2009): ACO can simulate 2-OPT.

Probability of particular 2-Opt step (for constant ρ):

MMAS$^*_{Ord}$: $\Theta(1/n^3)$ MMAS$^*_{Arb}$: $\Theta(1/n^2)$



Average Case Analysis

Assume that n points placed independently, uniformly at random in the unit hypercube $[0,1]^d$.

Theorem [Englert, Röglin, Vöcking 2007]

2-Opt finds after $O(n^{4+1/3} \cdot \log n)$ iterations with probability $1 - o(1)$ a solution with approximation ratio $O(1)$.

Theorem

For $\rho = 1$, MMAS$^*_{Arb}$ finds after $O(n^{6+2/3})$ iterations with probability $1 - o(1)$ a solution with approximation ratio $O(1)$.

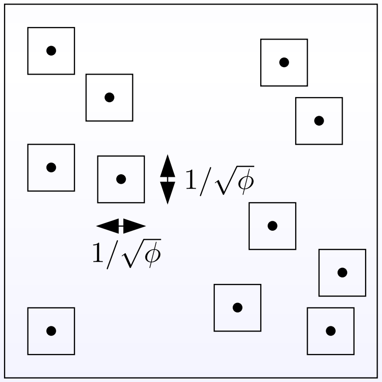
Theorem

For $\rho = 1$, MMAS$^*_{Ord}$ finds after $O(n^{7+2/3})$ iterations with probability $1 - o(1)$ a solution with approximation ratio $O(1)$.

Smoothed Analysis

Smoothed Analysis

Each point $i \in \{1, \ldots, n\}$ is chosen independently according to a probability density $f_i : [0,1]^d \to [0, \phi]$.



2-Opt:
$O(\sqrt[d]{\phi})$-approximation in $O(n^{4+1/3} \cdot \log(n\phi) \cdot \phi^{8/3})$ steps

MMAS$^*_{Ord}$: $O(\sqrt[d]{\phi})$-approximation in $O(n^{7+2/3} \cdot \phi^3)$ steps

MMAS$^*_{Arb}$: $O(\sqrt[d]{\phi})$-approximation in $O(n^{6+2/3} \cdot \phi^3)$ steps

ACO: Summary and Open Questions

(Stochastic) Shortest Paths

Natural and interesting test-bed for the robustness of ACO algorithms.

- global pheromone updates?
- other strategies to deal with noise
- where does slow pheromone adaptation help?
- average-case analyses with heuristic information

Strength of ACO

Problem-specific construction procedures can make ACO more powerful.

- how to find a fruitful combination of metaheuristic search and problem-specific components?

Main Challenge in Analysis of ACO

Understand dynamics of pheromones within borders.

- results for MST and TSP with more natural pheromone models

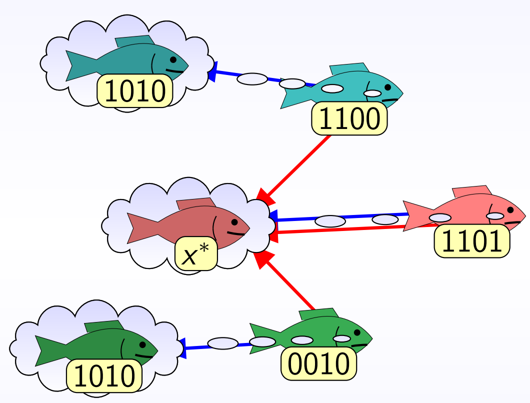
Overview

Particle Swarm Optimization

Particle Swarm Optimization

- Bio-inspired optimization principle developed by Kennedy and Eberhart (1995).
- Mostly applied in continuous spaces.
- Swarm of particles, each moving with its own velocity.
- Velocity is updated according to
  - own best position and
  - position of the best individual in its neighborhood (here: swarm).

Particle Swarm Optimization

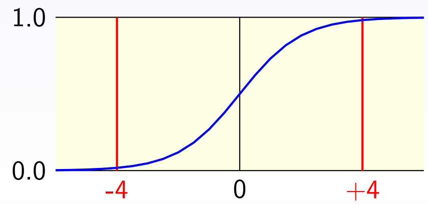


Binary PSO (Kennedy und Eberhart, 1997)

Creating New Positions

Probabilistic construction using velocity v and sigmoid function $s(v)$:

$$\text{Prob}(x_j = 1) = s(v_j) = \frac{1}{1+e^{-v_j}}$$



Restrict velocities to $v_j \in [-v_{\max}, +v_{\max}]$.

- Common practice: $v_{\max} = 4$ (good for $n \in [50, 500]$)
- Sudholt and Witt, 2010: $v_{\max} := \ln(n-1)$ (good across all n):

$$\frac{1}{n} \leq \text{Prob}(x_j = 1) \leq 1 - \frac{1}{n}.$$

Updating Velocities

Update current velocity vector according to

- cognitive component → towards own best: $x^{*(i)} - x^{(i)}$ and
- social component → towards global best: $x^* - x^{(i)}$.

Learning rates c_1, c_2 affect weights for the two components.

Random scalars $r_1 \in U[0, c_1]$, $r_2 \in U[0, c_2]$ chosen anew in each generation:

$$v^{(i)} = v^{(i)} + r_1(x^{*(i)} - x^{(i)}) + r_2(x^* - x^{(i)})$$

Understanding Velocities

Assume bit i is 1 in global best and own best. Create x.

- **ACO**: reinforce bit value 1 in probabilistic model if $x_i = 1$
- **PSO**: reinforce bit value 1 in probabilistic model if $x_i = 0$

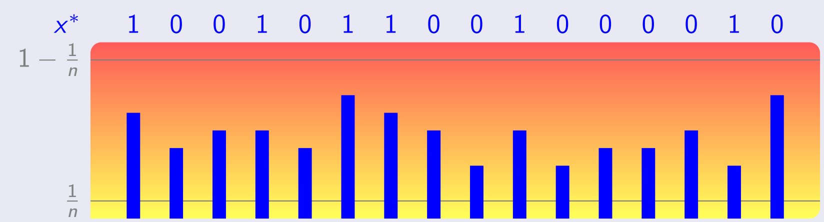
Probability to increase v_i is

$$1 - s(v_i) = s(-v_i) = \frac{1}{1+e^{v_i}}.$$

⇒ at least 1/2 for $v_i < 0$, but decreases rapidly with growing v_i.

Velocity Freezing

Particle with best-so-far solution: own best = global best

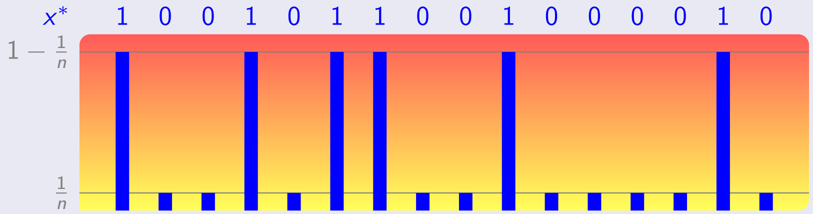


Lemma

Expected freezing time to $v_{\max}$ or $-v_{\max}$ is $O(n)$ for single bits and $O(n \log n)$ for n or μn bits if $\mu = poly(n)$.

Velocity Freezing

Particle with best-so-far solution: own best = global best



Lemma

Expected freezing time to $v_{\max}$ or $-v_{\max}$ is $O(n)$ for single bits and $O(n \log n)$ for n or μn bits if $\mu = poly(n)$.

Fitness-Level Method for Binary PSO

Upper bound for the (1+1) EA

$$\sum_{i=0}^{m-1} \frac{1}{s_i}$$

Upper bound for #generations of Binary PSO

$$\sum_{i=0}^{m-1} \frac{1}{s_i} + O(m \cdot n \log n)$$

Upper bound for #generations of "social" Binary PSO, i. e., $c_1 := 0$

$$O\left(\frac{1}{\mu} \sum_{i=0}^{m-1} \frac{1}{s_i} + m \cdot n \log n\right)$$

1-PSO vs. (1+1) EA on OneMax

More detailed analysis: average adaptation time of $384 \ln n$ is sufficient.

Theorem (Sudholt and Witt, 2010)

The expected optimization time of the 1-PSO *on* OneMax *is $O(n \log n)$.*

Proof uses layering argument and amortized analysis.

Experiments: 1-PSO 15% slower than (1+1) EA on OneMax.

Overview

Continuous PSO

Search space: (bounded subspace of) $\mathbb{R}^n$.

Objective function: $f\colon \mathbb{R}^n \to \mathbb{R}$.

Particles represent positions $x^{(i)}$ in this space.

Particles fly at certain velocity: $x^{(i)} := x^{(i)} + v^{(i)}$.

Velocity update with inertia weight ω:

$$v^{(i)} = \omega v^{(i)} + r_1(x^{*(i)} - x^{(i)}) + r_2(x^* - x^{(i)})$$

Convergence of PSO

Swarm can collapse to points or other low-dimensional subspaces.

Convergence results for standard PSO, $\omega < 1$ (Jiang, Luo, and Yang, 2007)

PSO converges ... somewhere.

Extensions of standard PSO

- Bare-bones PSO (Kennedy, 2003)
- PSO with mutation (several variants)
- PSO using gradient information (several variants)
- Guaranteed Convergence PSO (GCPSO) (van den Bergh and Engelbrecht, 2002)

Guaranteed Convergence PSO

Van den Bergh and Engelbrecht, 2002:

- Make a cube mutation of a particle's position by adding $p \in U[-\ell, \ell]^n$.
- Adapt "step size" ℓ in the course of the run by doubling or halving it, depending on the number of successes.

Possible step size adaptation (Witt, 2009)

After an observation phase consisting of n steps has elapsed, double ℓ if the total number of successes was at least $n/5$ in the phase and halve it otherwise. Then start a new phase.

$\longrightarrow$ 1/5-rule known from evolution strategies!

GCPSO with 1 Particle

GCPSO with one particle is basically a (1+1) ES with cube mutation.

Can be analyzed like classical (1+1) ES (Jägersküpper, 2007)

$$\textsc{Sphere}(x) := ||x|| = x_1^2 + x_2^2 + \cdots + x_n^2$$

Theorem (Witt, 2009)

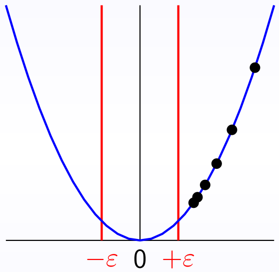
Consider the GCPSO$_1$ on $\textsc{Sphere}$. *If* $\ell = \Theta(||x^*||/n)$ *for the initial solution* x^**, the runtime until the distance to the optimum is no more than* $\varepsilon||x^*||$ *is* $O(n\log(1/\varepsilon))$ *with probability at least* $1 - 2^{-\Omega(n)}$ *provided that* $2^{-n^{O(1)}} \le \varepsilon \le 1$.

Same result as for (1+1) ES using Gaussian mutations in Jägersküpper, 2007.

Stagnation of Standard PSO

Lehre and Witt, MIC 2011

Standard PSO with one/two particles stagnates even on one-dimensional Sphere!



Expected first hitting time of ε-ball around optimum is infinite.

Noisy PSO (Lehre and Witt, 2011)

Adding noise $U[-\delta/2, \delta/2]$ for $\delta \leq \varepsilon$ yields finite expected first hitting time on (half-)Sphere.

PSO: Summary and Open Questions

Summary

- analysis of Binary PSO and its probabilistic model
- initial result on runtime of GCPSO
- results on expected first hitting time of ε-ball for Standard PSO & Noisy PSO

Neighborhood topologies

- ring topology, etc. instead of global best of swarm
- where does a restricted topology help?

Swarm dynamics

- analyze combined impact of cognitive and social components
- more results on swarms in continuous spaces

Overview

Conclusions

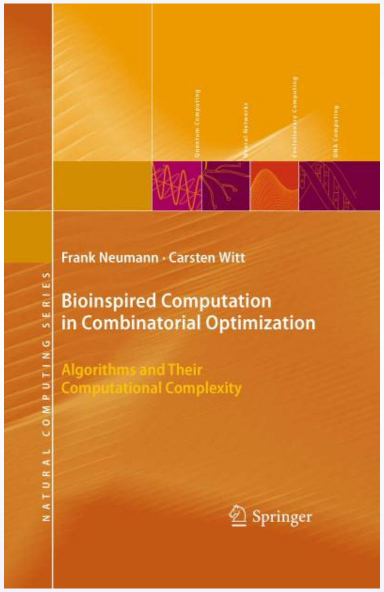
Summary

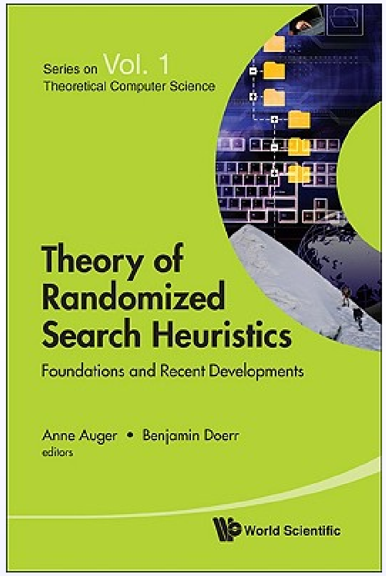
- Insight into probabilistic models underlying ACO and PSO
- How design choices and parameters affect (bounds on) running times
- How simple ACO algorithms optimize unimodal functions and plateaus
- Results for ACO in combinatorial optimization
- First analyses of basic PSO algorithms in discrete and continuous spaces

Future Work

- A unified theory of randomized search heuristics?
- More results on multimodal problems
- When and how diversity and slow adaptation help

Further Reading





Selected Literature I

Conference/workshop papers superseded by journal papers are omitted.
Preliminary versions of "Sudholt and Thyssen" (né Horoba) appeared as "Horoba and Sudholt".

N. Attiratanasunthron and J. Fakcharoenphol.
A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs.
Information Processing Letters, 105(3):88–92, 2008.

B. Doerr, A. R. Hota, and T. Kötzing.
Ants easily solve stochastic shortest path problems.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2012)*, 2012.
To appear.

B. Doerr and D. Johannsen.
Refined runtime analysis of a basic ant colony optimization algorithm.
In *Proceedings of the Congress of Evolutionary Computation (CEC '07)*, pages 501–507. IEEE Press, 2007.

B. Doerr, D. Johannsen, and C. H. Tang.
How single ant ACO systems optimize pseudo-Boolean functions.
In *Parallel Problem Solving from Nature (PPSN X)*, pages 378–388. Springer, 2008.

B. Doerr, F. Neumann, D. Sudholt, and C. Witt.
Runtime analysis of the 1-ANT ant colony optimizer.
Theoretical Computer Science, 412(17):1629–1644, 2011.

M. Dorigo and C. Blum.
Ant colony optimization theory: A survey.
Theoretical Computer Science, 344:243–278, 2005.

M. Dorigo and T. Stützle.
Ant Colony Optimization.
MIT Press, 2004.

Selected Literature II

W. J. Gutjahr.
On the finite-time dynamics of ant colony optimization.
Methodology and Computing in Applied Probability, 8:105–133, 2006.

W. J. Gutjahr.
Mathematical runtime analysis of ACO algorithms: Survey on an emerging issue.
Swarm Intelligence, 1:59–79, 2007.

W. J. Gutjahr.
First steps to the runtime complexity analysis of ant colony optimization.
Computers and Operations Research, 35(9):2711–2727, 2008.

W. J. Gutjahr.
Ant colony optimization: recent developments in theoretical analysis.
In *Theory of Randomized Search Heuristics–Foundations and Recent Developments*. World Scientific Publishing, 2011.

W. J. Gutjahr and G. Sebastiani.
Runtime analysis of ant colony optimization with best-so-far reinforcement.
Methodology and Computing in Applied Probability, 10:409–433, 2008.

Z. Hao, H. Huang, X. Zhang, and K. Tu.
A time complexity analysis of ACO for linear functions.
In *Simulated Evolution and Learning (SEAL 2006)*, volume 4247 of *LNCS*, pages 513–520. Springer, 2006.

J. Kennedy and R. C. Eberhart.
Particle swarm optimization.
In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948. IEEE Press, 1995.

J. Kennedy and R. C. Eberhart.
A discrete binary version of the particle swarm algorithm.
In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (WMSCI)*, pages 4104–4109, 1997.

Selected Literature III

J. Kennedy, R. C. Eberhart, and Y. Shi.
Swarm Intelligence.
Morgan Kaufmann, 2001.

T. Kötzing, P. K. Lehre, P. S. Oliveto, and F. Neumann.
Ant colony optimization and the minimum cut problem.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 1393–1400. ACM, 2010.

T. Kötzing, F. Neumann, H. Röglin, and C. Witt.
Theoretical analysis of two ACO approaches for the traveling salesman problem.
Swarm Intelligence, 6:1–21, 2012.

T. Kötzing, F. Neumann, D. Sudholt, and M. Wagner.
Simple Max-Min ant systems and the optimization of linear pseudo-Boolean functions.
In *Proceedings of the 11th Workshop on Foundations of Genetic Algorithms (FOGA 2011)*, pages 209–218. ACM Press, 2011.

P. K. Lehre and C. Witt.
Finite first hitting time versus stochastic convergence in particle swarm optimisation.
In *Proceedings of the 9th Metaheuristics International Conference (MIC 2011)*, 2012.
To appear.

F. Neumann, D. Sudholt, and C. Witt.
Rigorous analyses for the combination of ant colony optimization and local search.
In *Proceedings of the Sixth International Conference on Ant Colony Optimization and Swarm Intelligence (ANTS '08)*, volume 5217 of *LNCS*, pages 132–143. Springer, 2008.

F. Neumann, D. Sudholt, and C. Witt.
Analysis of different MMAS ACO algorithms on unimodal functions and plateaus.
Swarm Intelligence, 3(1):35–68, 2009.

Selected Literature IV

F. Neumann, D. Sudholt, and C. Witt.
Computational complexity of ant colony optimization and its hybridization with local search.
In C. P. Lim, L. C. Jain, and S. Dehuri, editors, *Innovations in Swarm Intelligence*, number 248 in SGI. Springer, 2009.

F. Neumann, D. Sudholt, and C. Witt.
A few ants are enough: ACO with iteration-best update.
In *Genetic and Evolutionary Computation Conference (GECCO '10)*, pages 63–70, 2010.

F. Neumann and C. Witt.
Runtime analysis of a simple ant colony optimization algorithm.
Algorithmica, 54(2):243–255, 2009.

F. Neumann and C. Witt.
Ant colony optimization and the minimum spanning tree problem.
Theoretical Computer Science, 411(25):2406 – 2413, 2010.

T. Stützle and H. H. Hoos.
MAX-MIN ant system.
Journal of Future Generation Computer Systems, 16:889–914, 2000.

D. Sudholt.
Using Markov-chain mixing time estimates for the analysis of ant colony optimization.
In *Proceedings of the 11th Workshop on Foundations of Genetic Algorithms (FOGA 2011)*, pages 139–150. ACM Press, 2011.

D. Sudholt and C. Thyssen.
Running time analysis of ant colony optimization for shortest path problems.
Journal of Discrete Algorithms, 10:165–180, 2012.

Selected Literature V

D. Sudholt and C. Thyssen.
A simple ant colony optimizer for stochastic shortest path problems.
Algorithmica, 2012.
To appear.

D. Sudholt and C. Witt.
Runtime analysis of a binary particle swarm optimizer.
Theoretical Computer Science, 411(21):2084–2100, 2010.

C. Witt.
Rigorous runtime analysis of swarm intelligence algorithms - an overview.
In *Swarm Intelligence for Multi-objective Problems in Data Mining*, number 242 in Studies in Computational Intelligence (SCI), pages 157–177. Springer, 2009.

C. Witt.
Why standard particle swarm optimisers elude a theoretical runtime analysis.
In *Foundations of Genetic Algorithms 10 (FOGA '09)*, pages 13–20. ACM Press, 2009.

C. Witt.
Theory of particle swarm optimization.
In *Theory of Randomized Search Heuristics–Foundations and Recent Developments*. World Scientific Publishing, 2011.

Y. Zhou.
Runtime analysis of an ant colony optimization algorithm for TSP instances.
IEEE Transactions on Evolutionary Computation, 13(5):1083–1092, 2009.

Thank you!

Questions?