technische universität dortmund

Computational Intelligence

Winter Term 2022/23

Prof. Dr. Günter Rudolph
Lehrstuhl für Algorithm Engineering (LS 11)
Fakultät für Informatik
TU Dortmund

Contents

- Ant algorithms (combinatorial optimization)
- Particle swarm algorithms (optimization in $\mathbb{R}^n$)

metaphor

swarms of bird or fish seeking for food	ants or termites seeking for food
concepts: • evaluation of own current situation • comparison with other conspecific • imitation of behavior of successful conspecifics	concepts: • communication / coordination by means of „stigmergy" • reinforcement learning → positive feedback
⇒ audio-visual communication	⇒ olfactoric communication

ant algorithms (ACO: Ant Colony Optimization)

paradigm for design of metaheuristics for combinatorial optimization

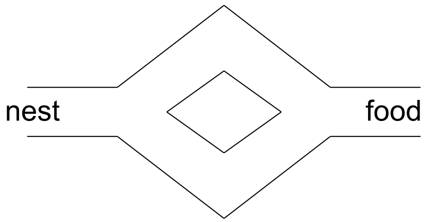
stigmergy = indirect communication through modification of environment

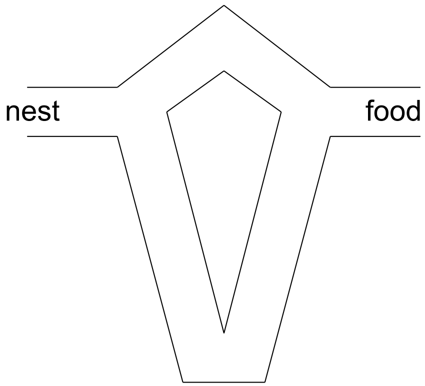
» 1991 Colorni / Dorigo / Maniezzo: Ant System (also: 1st ECAL, Paris 1991)
Dorigo (1992): collective behavor of social insects (PhD)

some facts: https://doi.org/10.1073/pnas.2201550119 (from 2022)

- about 2% of all insects are social
- about 50% of all social insects are ants
- total weight of all ants = 20% of weight of all humans
- ants populate earth since > 100 millions years (as old as dinosaurs!)
- *homo sapiens* populate earth since about 300,000 years (earlier versions extinct)

double bridge experiment (Deneubourg et al. 1990, Goss et al. 1989)





initially:
both bridges used equally often

finally:
all ants run over single bridge only!

finally:
all ants use the shorter bridge!

How does it work?

- ants place pheromons on their way
- routing depends on concentration of pheromons

more detailed:

ants that use shorter bridge return faster

→ pheromone concentration higher on shorter bridge

→ ants choose shorter bridge more frequently than longer bridge

→ pheromon concentration on shorter bridge even higher

→ even more ants choose shorter bridge

→ a.s.f.

positive feedback loop

Ant System (AS) 1991

combinatorial problem:

- components $C = \{ c_1, c_2, \ldots, c_n \}$
- feasible set $F \subseteq 2^C$
- objective function $f: 2^C \to \mathbb{R}$

ants = set of concurrent (or parallel) asynchronous agents move through <u>state of problems</u>

↓

partial solutions of problems

→ caused by movement of ants the final solution is compiled incrementally

movement: stochastic local decision

(2 parameters)

↙	↘
‘trails‘	‘attractiveness‘
paths	excitement, stimulus

while constructing the solution (if possible), otherwise at the end:

1. evaluation of solutions
2. modification of ‘trail value‘ of components on the path

$\underbrace{\text{evaluation / modification}}_{\text{feedback}}$

ant k in state i

- determine all possible continuations of current state i
- choice of continuation according to probability distribution p_{ij}

p_{ij} = q(attractivity, amount of pheromone)

heuristic is based on *a priori* desirability of the move

a posteriori desirability of the move „how rewarding was the move in the past?"

- update of pheromone amount on the paths:
  as soon as all ants have compiled their solutions
  good solution ↑ increase amount of pheromone, otherwise decrease ↓

Combinatorial Problems (Example TSP)

TSP:

- ant starts in arbitrary city i
- pheromone on edges (i, j): τ_{ij}
- probability to move from i to j: $p_{ij}^{(t)} = \dfrac{\tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}}{\sum\limits_{k \in \mathcal{N}_i(t)} \tau_{ik}^{\alpha}\,\eta_{ik}^{\beta}}$ for $j \in \mathcal{N}_i(t)$
- $\eta_{ij} = 1/d_{ij}$; d_{ij} = distance between city i and j
- $\alpha = 1$ and $\beta \in [2, 5]$ (empirical), $\rho \in (0,1)$ "evaporation rate"
- $\mathcal{N}_i(t)$ = neighborhood of i at time step t (without cities already visited)
- update of pheromone after μ journeys of ants: $\tau_{ij} := \rho\,\tau_{ij} + \sum\limits_{k=1}^{\mu} \Delta\tau_{ij}(k)$
- $\Delta\tau_{ij}(k)$ = 1 / (tour length of ant k), if (i,j) belongs to tour

two additional mechanisms:

1. *trail evaporation*
2. *demon actions* (for centralized actions; not executable in general)

Ant System (AS) is prototype

tested on TSP-Benchmark → not competitive

→ but: works in principle!

subsequent: 2 targets

1. increase efficiency (→ competitiveness with *state-of-the-art* method)
2. better explanation of behavior

1995 ANT-Q (Gambardella & Dorigo), simplified: 1996 ACS *ant colony system*

Particle Swarm Optimization (PSO)

↓

abstraction from fish / bird / bee swarm

paradigm for design of metaheuristics for continuous optimization

developed by Russel Eberhard & James Kennedy (~1995)

concepts:

- particle (x, v) consists of position $x \in \mathbb{R}^n$ and "velocity" (i.e. direction) $v \in \mathbb{R}^n$
- PSO maintains multiple potential solutions at one time
- during each iteration, each solution/position is evaluated by an objective function
- particles "fly" or "swarm" through the search space
  to find position of an extremal value returned by the objective function

PSO update of particle (x_i, v_i) at iteration t

1st step:

$$v_i(t+1) = \omega\, v_i(t) + \gamma_1\, R_1\,(x_b^*(t) - x_i(t)) + \gamma_2\, R_2\,(x^*(t) - x_i(t))$$

const. const. const.

random variable random variable

best solution among all solutions of iteration t ≥ 0

best solution among all solutions up to iteration t ≥ 0

$$x_b^*(t) = \underset{i=1,\ldots,\mu}{\mathrm{argmin}}\{f(x_i(t))\} \qquad x^*(t) = \underset{\tau=0,\ldots,t}{\mathrm{argmin}}\{f(x_b^*(\tau))\}$$

PSO update of particle (x_i, v_i) at iteration t

1st step:

$$\underbrace{v_i(t+1)}_{\text{new direction}} = \omega\, \underbrace{v_i(t)}_{\text{old direction}} + \gamma_1\, R_1\, \underbrace{(x_b^*(t) - x_i(t))}_{\text{direction from } x_i(t) \text{ to } x_b^*(t)} + \gamma_2\, R_2\, \underbrace{(x^*(t) - x_i(t))}_{\text{direction from } x_i(t) \text{ to } x^*(t)}$$

ω	:	inertia factor, often $\in [0.8, 1.2]$
γ_1	:	cognitive factor, often $\in [1.7, 2.0]$
γ_2	:	social factor, often $\in [1.7, 2.0]$
R_1	:	positive r.v., often $r_1 \sim U[0,1]$
R_2	:	positive r.v., often $r_2 \sim U[0,1]$

PSO update of particle (x_i, v_i) at iteration t

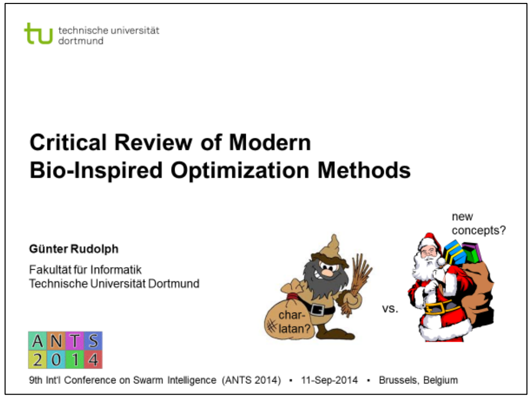
2nd step:

$$\underbrace{x_i(t+1)}_{\text{new position}} = \underbrace{x_i(t)}_{\text{old position}} + \underbrace{v_i(t+1)}_{\text{new direction}}$$

Note the similarity to the concept of mutative step size control in EAs: first change the step size (direction), then use changed step size (direction) for changing position.

More swarm algorithms:

- Artificial Bee Colony
- Krill Herd Algorithm
- Firefly Algorithm
- Glowworm Swarm
- ...



First presented 06-Dec-2013 @ CI Workshop Dortmund

But be watchful:

Is there a new algorithmic idea inspired from the biological system?

Take a look at the code / formulas: Discover similarities & differences!

Often: “Old wine in new skins.“

Biological Original

Firefly Algorithm (FA)

firefly / glowworm / glowfly …

nocturnal animal



is able to emit light at abdomen by chemical reaction;

specific luminescent rhythm per species → find mating partner

→ warns other species (uneatable)

→ decoys other species (as food source)

<u>assumptions:</u>

- unisex ⇒ symmetric attractiveness between mating partners
- attractiveness proportional to luminance;
- luminance proportional to objective function value (if → max!)
- movement in direction of light source
- light intensity decreases as distance increases
- if no mating partner in vicinity (i.e., not visible), then random movement

Implementation: Pseudo Code (→ max!)

Firefly Algorithm (FA)

initialize μ fireflies $x_1, \ldots x_\mu \in R^n$ with light intensity $f(x_1), \ldots, f(x_\mu)$
repeat
  foreach x:
   foreach y ≠ x:
(a)   **if** f(y) > f(x) **then**
     fly from x toward y:
     $x = x + \underbrace{\beta \exp(-\gamma \, \| y - x \|^2)}_{\text{attractiveness}} (y - x) + \underbrace{U_n[-\alpha, \alpha] / 2}_{\text{„mutation“}}$
    endif
   endfor
  endfor
  store the best firefly
until happy

recommended:

$\alpha \in [0,1]$

$\beta = 1$

$\gamma = 1$

(a) **What happens actually?** → alternative formulation:

sort fireflies $f(x_{1:\mu}) \leq f(x_{2:\mu}) \leq \ldots \leq f(x_{\mu:\mu})$

$$x_{i:\mu} = x_{i:\mu} + \underbrace{\sum_{j=i+1}^{\mu} (x_{j:n} - x_{i:n}) \cdot w(\|x_{j:n} - x_{i:n}\|)}_{\text{weighted recombination?}} + \underbrace{\frac{1}{2} \sum_{j=i+1}^{\mu} U_n[-\alpha, \alpha]}_{\text{mutation!}}$$

Implementation: Analysis

Firefly Algorithm (FA)

$$\begin{aligned} x_{i:\mu} &= x_{i:\mu} + \sum_{j=i+1}^{\mu} (x_{j:n} - x_{i:n}) \cdot w_{ij} + \frac{1}{2} \sum_{j=i+1}^{\mu} U_n[-\alpha, \alpha] \\ &= \underbrace{\left(1 - \sum_{j=i+1}^{\mu} w_{ij}\right) \cdot x_{i:\mu} + \sum_{j=i+1}^{\mu} w_{ij} \cdot x_{j:\mu}}_{\Rightarrow \text{ weighted multi - recombination}} + \underbrace{\frac{1}{2} \sum_{j=i+1}^{\mu} U_n[-\alpha, \alpha]}_{\text{approximately Gaussian}} \end{aligned}$$

$$\sim N\left(0, \frac{(\mu - i)\,\alpha^2}{12} \cdot I_n\right)$$

new concepts?

• generates new points by weighted recombination and mutation (= EA)

• **but**: weighting depends on distance between individuals (inspired by original)

no benchmark results! (defines 2 own test problems for n=2 and n=5)

Biological Original

Bat Algorithm (BA)

bats (~ 1000 species, since 50 x 10^6 years)

3.3 cm body	14 cm body
17 cm wing-span	60 cm wing-span

essential distinctive feature: shape of head →

capable of **echo location**: (not all species!)

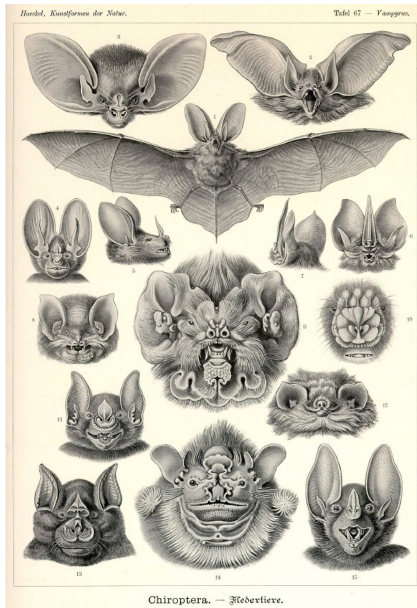
emits short ultra sonic impulses (mouth/nose),

radiation focusable

if prey reflects sound) # impulses ↑

noise loudness (ultrasonic) > jackhammer

frequency 25 – 150 kHz, pulse rate 10 – 20 Hz



https://doi.org/10.11588/diglit.3064#0061

Implementation

<u>assumptions:</u>

- localization and distinction between prey and obstacle by echo localization
- flight velocity v at position x
- search frequency $\in [\, F_{min} , F_{max} \,]$ (→ higher frequency has more energy?)
- loudness $\in [\, A_{min}$, A∈$_{max}$] decreases while approaching prey (→ why?)
- puls rate r > 0 increases while approaching prey (→ more precise localization)

<u>description of algorithm:</u>

- pseudo code in original paper extremly vague; verbal description unclear
- MATLAB code in monograph [Yang 2010] without loudness and puls rate
- MATLAB code in Matlab Central (July 2012) with different initialization
  http://www.mathworks.com/matlabcentral/fileexchange/37582-bat-algorithm-demo

Implementation: Pseudo Code

extracted from MATLAB code

initialize μ bats b = (x, v, F, r, A) and determine b* with best fitness value
$x_{best} = x^*$ and $f_{best} = f(x_{best})$
repeat
  for each bat
    $f_{old} = f(x)$
(a) $v = v + (x - x^*) \cdot U[\, F_{min}, F_{max} \,]$
    $x = x + v$
(b)   **if** U[0,1] > r **then** $x = x^* + \sigma \cdot N(0, I_n)$ mit σ 2 [-1, 1]
(c)   **if** $f(x) \le f_{old}$ **and** U[0,1] < A **then** accept new bat // copies only x !
    if $f(x) \le f_{best}$ **then** $x_{best} = x$; $f_{best} = f(x_{best})$
  endfor
until $| f_{best} - f_{opt} | < \varepsilon$

(a) search frequency has no strategic meaning (= random number).
(b) realizes an iteration of (1+1)-EA with probability 1 – r, fixed step size!
(c) an improvement (!) is accepted with prob. A , only x is copied!

Implementation: Pseudo Code

extracted from MATLAB code

initialize μ bats b = (x, v, F, r, A) and determine b* with best fitness value
$x_{best} = x^*$ and $f_{best} = f(x_{best})$
repeat
  for each bat
    $f_{old} = f(x)$
    $v = v + (x - x^*) \cdot U[\, F_{min}, F_{max} \,]$
    $x = x + v$
(b)   **if** U[0,1] > r **then** $x = x^* + \sigma \cdot N(0, I_n)$ mit $\sigma \in [-1, 1]$
(c)   **if** $f(x) \le f_{old}$ **and** U[0,1] < A **then** accept new bat // copies only x !
    if $f(x) \le f_{best}$ **then** $x_{best} = x$; $f_{best} = f(x_{best})$
  endfor
until $| f_{best} - f_{opt} | < \varepsilon$

original paper claims to realize following code snippet if bat improved:

(b) $r = r_{const} \, (1 - \exp(-\gamma \cdot t)) \to r_{const} > 0$ für $t \to 1$ $\Rightarrow \mu\,(1 - r_{const})$ (1+1)-steps
(c) $A = \alpha \cdot A$ mit $0 < \alpha < 1$ $\Rightarrow$ P{ accept better bat } → 0

Published Performance

<u>setting of experiments:</u>

GA = standard GA ($\mu = 40$, $p_m = 0.05$, $p_c = 0.95$),
PSO = standard PSO ($\mu = 40$, $\alpha = \beta = 2$, inertia = 1),
BA = BA with A_t ($\mu = 25..50$, $\alpha = 0.9$),

100 runs; mean #FEs Std.Dev. (success prob.) until $| f_{best} - f_{opt} | < \varepsilon = 10^{-5}$

Functions/Algorithms	GA	PSO	BA
Multiple peaks	$52124 \pm 3277 (98\%)$	$3719 \pm 205 (97\%)$	$1152 \pm 245 (100\%)$
Michalewicz's (d=16)	$89325 \pm 7914 (95\%)$	$6922 \pm 537 (98\%)$	$4752 \pm 753 (100\%)$
Rosenbrock's (d=16)	$55723 \pm 8901 (90\%)$	$32756 \pm 5325 (98\%)$	$7923 \pm 3293 (100\%)$
De Jong's (d=256)	$25412 \pm 1237 (100\%)$	$17040 \pm 1123 (100\%)$	$5273 \pm 490 (100\%)$
Schwefel's (d=128)	$227329 \pm 7572 (95\%)$	$14522 \pm 1275 (97\%)$	$8929 \pm 729 (99\%)$
Ackley's (d=128)	$32720 \pm 3327 (90\%)$	$23407 \pm 4325 (92\%)$	$6933 \pm 2317 (100\%)$
Rastrigin's	$110523 \pm 5199 (77\%)$	$79491 \pm 3715 (90\%)$	$12573 \pm 3372 (100\%)$
Easom's	$19239 \pm 3307 (92\%)$	$17273 \pm 2929 (90\%)$	$7532 \pm 1702 (99\%)$
Griewangk's	$70925 \pm 7652 (90\%)$	$55970 \pm 4223 (92\%)$	$9792 \pm 4732 (100\%)$
Shubert's (18 minima)	$54077 \pm 4997 (89\%)$	$23992 \pm 3755 (92\%)$	$11925 \pm 4049 (100\%)$

[Yang 2010, S. 73]

but: results with MATLAB-Code from `mathworks` <u>**not reproduceable**</u>!

Cause Study — Bat Algorithm (BA)

Xin-She Yang: *Nature-Inspired Metaheuristic Algorithms*, Luniver Press 2008
- contains MATLAB Code (seemingly transferred to 2nd ed. 2010 without change)
- contains no performance results

X.-S. Yang: *A New metaheuristic Bat-Inspired Algorithm,* S. 65-74, in NISCO 2010
- contains no MATLAB code (presumably used code from 2008/2010 monograph)
- contains performance results (see previous slide)

How did results materialize?

→ “convenient“ initialization [Yang 2010, p. 102]

almost all test problems have optimal solution in origin;

all bats initialized via `x = randn(1, n)` → $x \sim N(0, I_n)$

thus, all bats standard normal-distributed around global optimum!

MATLAB code from `mathworks` initializes “correctly“ → bad results!

Upshot — Bat Algorithm (BA)

- specific properties of biological original not realized
- hence, no new concepts ⇒ PSO-like swarm variant with (1+1)-steps
- no reference code; only incomplete “demo“ versions
- performance not reproduceable!

⇒ **should never have been published!**

X.-S. Yang & S. Deb: Bat algorithm for multi-objective optimisation, *Int‘l J. Bio-Inspired Computation 3(5):267-274*, 2011.

→ same algorithm with scalarization via weighted sum!

What kind of journal is it? (Impact Factor 2012: 1.351)

EiC: Zhihua Cui

Advisory Board: X.-S. Yang, S. Deb, et al.

Be aware: journal published by **Inderscience**

<u>not:</u> Interscience (Wiley)

Biological Original — Cuckoo Algorithm (CA)

cuckoo (~ 130 species)

name rooted in birdcall „cuckoo“ of male

characteristic feature (of only ~ 40 species):
brood parasitism

- cuckoo places own eggs in foreign nests
- if event not noticed by host animal, cuckoo‘s egg will be incubated by host bird
- cuckoo hatches <u>first</u> (10-13 days) + kicks (all) other eggs out of nest
- mimicks call for foster mother
- grows faster than other birds

Implementation — Cuckoo Algorithm (CA)

<u>assumptions:</u>
- each cuckoo lays exactly 1 egg
- the “best“ nests with “high-qualitaty“ eggs will pass to next iteration
- host bird detects cuckoo‘s egg with prob. p (removes egg or build new nest)
- cuckoo = cuckoo‘s egg = nest

<u>description of algorithm:</u>
- pseudo code in original literature extremly vague; verbal description unclear
- MATLAB code in monograph [Yang 2010] with fixed parametrization
- deploys Gaussian instead of purported Lévy distribution
- no benchmarks! (claims verbally good results for Michalewicz test function n=2)

Pseudo Code (→ max!)

initialize μ nests/eggs $x_1, \ldots x_\mu \in R^n$ with egg quality $f(x_1), \ldots, f(x_\mu)$
repeat
  choose a cuckoo x_i at random
  cuckoo flies: $x = x_i + \alpha \cdot$ Lévy
  choose a nest x_j at random ($j \neq i$ not required)
  if $f(x) > f(x_j)$ **then** lay egg x into nest x_j // thus: $x_j = x$
  replace fraction p of worst nests with random new nests
  store best nest / cuckoo / egg
until happy

inspection of Matlab Code:

- random new nests are normally distributed around old nest $x = x + 0.01 \cdot \Delta \cdot N(0, I_n)$
- cuckoo flies normally distributed and not according to Lévy

What happens? May be interpreted as follows:

fraction of p individuals move as per random walk
fraction of 1-p individuals perform ($[1-p]\cdot\mu$ + 1)-EA with random replacement (no reco.)

Upshoot

- brood parasitism realized in no way!
- algorithm resembles poorly designed EA
  - with very weak selection pressure
  - with no recombination
  - with no step size control
  - with wasting of $p \cdot \mu$ FEs by Gaussian random walk

) **cannot be competitive!**

Where can this be published?

- Proceedings NaBic 2009: X.-S. Yang & S. Deb: Cuckoo Search via Lévy Flights
- X.-S. Yang & S. Deb: $\underbrace{\textit{Int. J. Math. Modelling \& Num. Opt.}}_{\text{Editor-in-Chief: Xin-She Yang}}$ *1:330-343*, 2010.

The Metaphor Crisis

There are hundreds of "animal/plant algorithms" (metaheuristics)

→ see the list 'EC Bestiary' @ http://fcampelo.github.io/EC-Bestiary/

Conjecture

Authors took a blind pick from any encyclopedia of animals or plants

to weirdly describe an algorithm that is purportedly inspired by that species.

Be alerted!

If you see a "new" bio-inspired algorithm, ask

- what are the properties of biological original?
- which assumptions/simplifications have been made?
- which properties have been implemented?
- are new concepts for optimization identifiable?
- motto: „deflate verbal bubbles" – inspect the formulas!
- how did they compare algorithms' performance?

The Metaphor Crisis

May it be hoax, ignorance, fraud, naivity of the authors …

The CI community must fight against the metaphor glut,

as these publications can be harmful to the reputation of CI!

Journals

Journal of Heuristics (2015)
Swarm Intelligence (2016),
ACM Transactions on Evolutionary Learning and Optimization (2021)

have additions to their submission guidelines:

This journal will not publish papers that propose "novel" metaphor-based metaheuristics, unless the authors
(i) present their method using the normal, standard optimization terminology;
(ii) show that the new method brings useful and novel concepts to the field;
(iii) motivate the use of the metaphor on a sound, scientific basis; and
(iv) present a fair comparison with other state-of-the-art methods using state-of-the-art practices for benchmarking algorithms.