Course: Natural Computing

5. Metaheuristic Optimisation Algorithms

J. Michael Herrmann
School of Informatics, University of Edinburgh
michael.herrmann@ed.ac.uk, +44 131 6 517177

Overview

1. No free lunch theorem (previously)
2. Model based search (previously)
3. Random walks, informed search, Bayesian estimation (now)
4. Biologically and physically inspired algorithms (now)
5. Schema theory for GA & GP (soon)
6. Dynamics and convergence (soon)

1. What happens in MHO?

Because of the diversity of the "inspirations" of MHO algorithm, a common mechanism is hardly visible. The common theme seems to be to achieve either a balance between to opposed effects or a gradual shift between them:

- Exploration vs. exploitation
- Cooperation vs. competition
- Diversification vs. intensification (Blum and Roli, 2003)
- Global search vs. local descent/ascent
- Randomness vs. greediness (goal-directedness)

These concepts set the scene, but what happens?

Exploration and exploitation in our algorithms

	exploration	exploitation
SA	temperature-based fitness decrease	fitness increase (hill-climbing)
GA, ES	mutation crossover (islands)	selection (elitism)
ACO	probability rule (sampling) $\tau_{\min}$, $\tau_{\max}$	pheromone evaporation, local heuristics (local search)
PSO	inertia (ω) overshooting forces ($\alpha_1 + \alpha_2 > \approx 4$)	forces to bests ($\alpha_1 + \alpha_2 < \approx 4$), "constriction"

How can exploration and exploitation be balanced or controlled?

Exploration and exploitation

"A metaheuristic will be successful on a given optimization problem if it can provide a balance between the exploitation of the accumulated search experience and the exploration of the search space to identify regions with high quality solutions in a problem specific, near optimal way."

T. Stuetzle: *Local Search Algorithms for Combinatorial Problems—Analysis, Algorithms and New Applications.* DISKI. infix, St. Augustin, 1999.

Random walks

- Undirected, local search

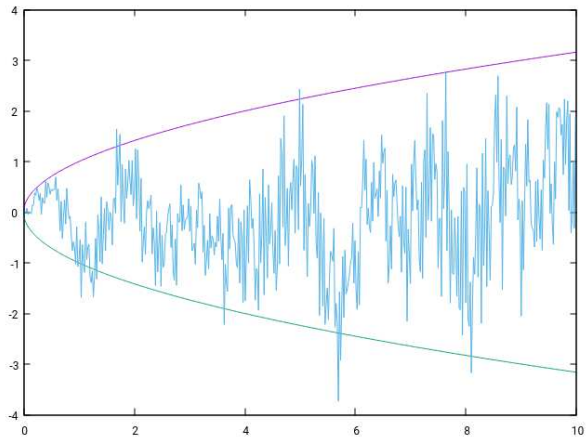
$$x \leftarrow x + \Delta x$$

  where Δx is a random vector.
- Population size 1
- Can be "global" or "local" in dependence on average step width
- Isotropy and independence of increments are often assumed in modelling, but are not required in a general sense nor provided by all algorithms

Random walks: Example

- Discrete state space $\mathbb{Z}^d$, start at origin
- At each time step $t \in \mathbb{N}$ at with probability $\frac{1}{2}$ either $+1$ or -1

- For $d = 1$ and large t, the ensemble of the walks will approach a Gaussian distribution with mean $\mu = 0$ and variance $\sigma^2 \sim t$, i.e. the trajectory will stay with high probability within $\left[-\kappa\sqrt{t}, \kappa\sqrt{t}\right]$ for fixed κ, e.g. $p = 0.97$ per step for $\kappa = 3$.



- A random walk comes arbitrarily close to a global optimum x^* ($\|x^*\| < \infty$) with prob. 1 for $d \leq 2$, but not for $d > 3$, i.e. in higher dimensions we need to use additional information

Random walks

- Occur as special cases (e.g. for constant fitness function in many MHO algorithms)
- Are used in many MHO algorithms to introduce diversity
- Unbiased random walks can be used to find out about characteristics of the problem, i.e. the fitness landscape and thus help to avoid confounding properties of the problem and properties of the algorithm (e.g. consider random walk correlation function [Weinberger, 1990])
- Can serve as 0th-level comparisons for MHO algorithms
- Occur widely in other context, e.g. in reinforcement learning, or in biology (*E. coli* or viruses).

see e.g. Barry D. Hughes: *Random Walks and Random Environments* (1995)

Lévy flights

- A random walk with a step length the follows a Lévy distribution

$$P_{\text{Lévy}}(x) \sim a x^{-\gamma}, \text{ with } x > 0 \text{ and } \gamma \in (1, 3]$$

- Mean and variance are infinite ("heavy tail")
- Usually a lower and upper cut-off is used, i.e. $x \in [x_{\text{min}}, x_{\text{max}}]$ with $x_{\text{min}} > 0$, $x_{\text{max}} < \infty$.
- Compromise between local exploration and "quite a few" large-scale steps.
- Used in many MHO algorithms for *scale-free* search

Informed search

- Biased random walks
- A heuristic uses the information obtained by an optimisation algorithm in order to decide which candidate solutions will be tested in future.
- Heuristics depend on the type of problem.
- This includes also AI search methods such as best-first search
- Adaptive walks, i.e. walks that change their statistics based on a fitness function, are the link to MHO algorithms

Judea Pearl: *Heuristics* (1984); Michalewicz & Fogel: *How to Solve It* (2004)

Bayesian framework

- Start with a probability distribution that covers all the search space and which will be used as a prior.
- Fitness values F_i for points x_i obtained by time step t serve as evidence for an posterior distribution

$$P_t\left(x|x_i, F_i\right) = \frac{P\left(x|x_i, F_i\right) P_t\left(x\right)}{P\left(x_i, F_i\right)}$$

  which can be seen as expressing a belief about the global optimum.
- The posterior can be used as new prior $P_{t+1}\left(x\right) = P_t\left(x|x_i, F_i\right)$
- The sequence of $P_t\left(x\right)$, $t \geq 0$ may contract quickly, such is may be useful to inject additional entropy.

Biologically and Physically Inspired Algorithms

Course: Natural Computing (week 5)

J. Michael Herrmann
School of Informatics, University of Edinburgh
michael.herrmann@ed.ac.uk, +44 131 6 517177

Overview

- These additional algorithms are included here as a toolbox where bits and pieces can be repurposed for use in algorithms of your own design.
- The biology, physics, chemistry that provides the background (bat, bees, monkeys, ...) is certainly interesting in itself, but will not be included here (see Sörensen, 2013).

Continuous and combinatorial optimisation

- Combinatorial optimisation
  - exhaustive
  - polynomial time algorithms, e.g. Dijkstra's algorithm
  - approximations, e.g. metaheuristics:
    - Taboo search
    - Simulated annealing
    - Genetic algorithms
    - Ant colony optimisation (River formation dynamics)
- Continuous optimisation
  - local: gradients, downhill-simplex (Nelder-Mead method)
  - global: metaheuristics
    - PSO, ES, DE, ...

Preview

Most biologically inspired algorithms are

- similar to PSO
  - swarms of solutions: vectors in $\mathbb{R}^d$
  - attraction to better solutions
  - often no control of dynamics (compared to PSO: $\alpha = 1$ and $\omega = 0$)
- similar to ES by adding structure to the population (also seen in PSO with topology)
- similar to ACO
  - probability rule for more exploration near best solutions
  - preference of fitter solutions lead to premature convergence, which is counteracted by randomness

The Bat Algorithm (Yang, 2010)

With probability τ_i

$\rho \sim U\left[\rho_{\min}, \rho_{\max}\right]$

$$v_i\left(t+1\right) = v_i\left(t\right) + \rho\left(g - x_i\left(t\right)\right)$$
$$x_i\left(t+1\right) = x_i\left(t\right) + v_i\left(t+1\right)$$

With probability $1 - \tau_i$

$\rho_L \sim U\left[-1, 1\right]^D$

$$x_i = x_k + \rho_L \left(\frac{1}{N} \sum_{j=1}^{N} L_j \right)$$

The Bat Algorithm

With probability τ_i

$\rho \sim U[\rho_{\min}, \rho_{\max}]$

$$v_i(t+1) = v_i(t) + \rho(g - x_i(t))$$
$$x_i(t+1) = x_i(t) + v_i(t+1)$$

If $\tau_i \to 1$, the dynamics is similar to PSO with parameter $\alpha_1 = 0$ (no personal best)

$\rho_{\min}$ has little effect on the performance, so we can set $\rho_{\min} = 0$

$\rho_{\max}$ can now be chosen as α_2 for PSO (or a bit more conservative because the bat algorithm has extra noise)

The Bat Algorithm: "Cooling" scheme for the noise

- The noise strength $\frac{1}{N}\sum_{i=1}^{N} L_i$ is initialised by $L_i = 0.5$ for all i and reduced in each step by multiplication with α

  If $\rho < L_i$ $(\rho \sim U[0,1])$ $\quad L_i = \alpha L_i$
  The noise strength $\frac{1}{N}\sum_{j=1}^{N} L_j$ decays as $\kappa\frac{1}{t}$ for $t \to \infty$, with a factor $\kappa \sim \frac{1+\alpha+4\sqrt{\alpha}}{6(1-\alpha)}$ (empirical), i.e. the noise decays slowly for all $\alpha \in (0,1)$
  Note that the noise is not related to the dimension of the search space such that the initial values for L_i need to be set appropriately.
- The noise events occur with probability τ_i for the individual i
  The update rule for τ is (possibly) $\tau_i = 1 - e^{-\gamma t}$, such that all τ approach quickly 1. It is possible that a slower approach was intended.

The Bat Algorithm

Comparison

- Largely similar to PSO
- Randomness is added with a slowly decaying frequency

Problems

- For good parameters PSO does not need extra randomness
- For sub-optimal parameters, extra randomness can help, but is hard to control w.r.t. strength and timescale

Artificial Bee Colonies (Karaboga, 2005)

- Similar to DE/PSO with choice of vectors similar to ACO
- Step 1: Given a set of current "bests" s_i, mutate each s_i by mixing with another random solution s_j

$$s_i(t+1) = s_i(t) + \rho_B (s_i(t) - s_j(t)) \tag{1}$$

  $\rho_B \sim U[-1,1]$, usually for only one coordinate at a time.
- Step 2: Choose from a set of current "bests" s_i with probability

$$p_i = \frac{q_i}{\sum_j q_j}$$

  where q_i is the fitness ("interestingness") of the solution s_i, and perform (1) again to explore more near the best solutions
- New s_i is accepted only if better, if a "best" rarely leads to improvements it is replaced by a random solution

Artificial Bee Colonies (ABC)

Comparison

- Similar to DE: Differences between random solutions are added
- Similar to PSO: The difference is multiplied by a random factor ($\alpha = 1$)
- Similar to ACO: Solutions with higher fitness are preferred
- Randomness by adding random points in the search space

Problems

- No parameters (as in PSO/DE) to control the dynamics of the solutions
- Randomly added solutions have low fitness and are thus unlikely to receive proper exploration
- Probability tends quickly to a single (local) optimum
- If no further improvement possible, the bees move on to other random locations after $e_{\max}$ trials, i.e. $e_{\max}$ limits exploitation

Cuckoo Search (Yang, 2009)

- Two modes of particle movement, **local** (similar to DE)

$$s_i(t+1) = s_i(t) + \alpha\, \rho \circ (s_j(t) - s_k(t))$$

with step width α and random factor $\rho \in \{0, 1\}$, i.e. only some dimensions are changed, and **global**

$$s_i(t+1) = s_i(t) + \alpha\, \boldsymbol{\xi}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_d)$ and $|\xi_i| \sim P_{\text{Levy}}$

- A fraction of bad solutions is abandoned, and random solutions are added.

Spider Monkey algorithm (Bansal et al., 2014)

- Hierarchical version of ABC, e.g. 40 individuals forming up to 5 subgroups (similar to ES)
- Similar to PSO, individuals are updated by random amounts
  - attraction to group best and attraction to or repulsion from random individual from other group
  - attraction to global best and attraction to or repulsion from random individual from other group
  - attraction to global best and repulsion from group best
- Global and local bests are updated, global best does hill-climbing
- Groups are combined or split with a certain probability

Glow-worm Algorithm (Krishnanand & Ghose, 2006)

- Compare own fitness to individuals within a neighbourhood
- Calculate probability of glow-worms moving towards each of these neighbours (similar to ACO)
- Choose neighbour and move towards it (similar to PSO)
- Update neighbourhood range to maintain a group-based competition even at crowding near local optimum
- While individuals are still exploring, they may have empty environments (neighbourhood range has a maximum)
- The Firefly Algorithm also updates the maximum neighbourhood range

Harmony Search (Geem et al., 2001)

- Similar to ES (or population-based SA) at zero temperature plus random search
- Choose a random "best" s_i and check whether a change of a component

$$s_{ij} = s_{ij} + U[-\rho, \rho]$$

  leads to an improvement
- Accept change when better than the worst over all "bests"
- With a small probability τ add a new random "best" chosen from a certain range $[l, u]^d$
- τ and ρ are decreased over time
- Problem: Set of "bests" contracts quickly to the region near best "best".

Harmony Search (Geem et al., 2001)

- "Weyland (2013) demonstrates convincingly, that Harmony Search (Geem, 2001) is nothing more than a special case of Evolution Strategies (Beyer, 2002) in which each of the concepts of Evolution Strategies has been relabeled from an evolution-inspired term to a term inspired by musicians playing together."
- "Even though the development of Evolution Strategies precedes that of Harmony Search by at least 30 years, the latter is proposed as an innovation and has by now attracted an impressively long list of follow-up research." [$\approx$ 10,000 citations]

From Sörensen (2013), following Scholarpedia "Metaheuristics"

Physics- and Chemistry-based Algorithms

Siddique and Adeli (2017)

- Theoretical Physics as a science is based on optimisation:
  Principle of least action (or principle of stationary action):
  A particle starting at point x_1 at time t_1 and reaching position x_2 at time t_2 follows a trajectory that is an extremum of the action integral. $\Rightarrow$ Nonlocal optimisation in space and time!
- Potential advantage: Physical systems are well understood w.r.t. dynamics, fluctuations, perturbations etc.
- Can we extract methods for computational optimisation?
  - Computational problem may not have a physics counterpart
  - Physical motion may not be effectively computable (e.g. chaotic behaviour)
  - Physics systems may be computationally complex, too
  - Physics systems may reside in a local optimum
  - Physics or chemistry may still be merely a metaphor ...
- SA and PSO inspired by thermodynamics & statistical physics

Electromagnetism-like optimisation (EMO) Birbil & Fang(2003)

- Coulomb's law: Force between two particles with charges q_i and q_j at resp. positions $x_i$and x_j

$$F_{ij} = k_e \frac{q_i \, q_j}{\left|x_i - x_j\right|^2} \left(x_i - x_j\right)$$

- For use in optimisation, determine charges according to fitness

$$q_i = \exp\left(\frac{\left(f\left(x_i\right) - f\left(x_{\text{best}}\right)\right) D}{\sum_{j=1}^{N} f\left(x_j\right) - f\left(x_{\text{best}}\right)}\right)$$

where D is the dimension of the search space and N the number of particles

Electromagnetism-like optimisation (EMO)

- The direction of the force is defined as towards the particle with higher fitness, i.e. a particle is attracted to the better particle and repelled from a worse one
- Particle i experiences the following force from particle j

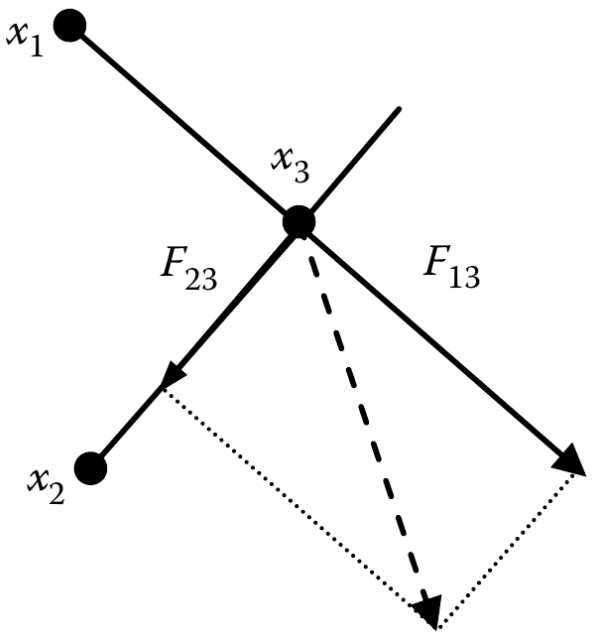
$$F_{ij} = \begin{cases} \frac{q_i\, q_j}{\left|x_i - x_j\right|^2} \left(x_j - x_i\right) & \text{attraction if } f\left(x_i\right) < f\left(x_j\right) \\ \frac{q_i\, q_j}{\left|x_i - x_j\right|^2} \left(x_i - x_j\right) & \text{repulsion if } f\left(x_i\right) \geq f\left(x_j\right) \end{cases}$$

- It moves then by a random amount ρ according to the force

$$x_i = x_i + \sum_{j \neq i} \frac{\rho F_{ij}}{|F_{ij}|}$$

- The best particle does not move.
- In spite of the normalisations, it is advisable to restrict the movement by bounds.

EMO: Function for 3 particles



x_3 is better than x_1,
but worse than x_2

- No natural scaling between forces (fitnesses) and states
- Spiralling-like dynamics,
- prone to premature convergence
- Repulsion strongest if both particles are not too bad
- Many variants that include more noise, subpopulations etc.
- Elsewhere, many nice applications of EM-like algorithms exists, e.g. untangling of graphs

Opposition-based EMO

- Given a population $X = \{x_1, \ldots, x_N\}$
- $\bar{x}_i = (\bar{x}_{i1}, \ldots, \bar{x}_{iD})$ is opposed to $x_i = (x_{i1}, \ldots, x_{iD})$ if $\bar{x}_{ik} = L_k + (U_k - x_{ik})$ for all k, where L_k and U_k are the lower and upper limits of the k-th dimension of the search space.
- Given X calculate $\bar{X} = \{\bar{x}_1, \ldots, \bar{x}_N\}$ and choose the N fittest individuals from $X \bigcup \bar{X}$
- Opposition procedure can be applied initially or also later during runtime, or only to dome of the dimensions
- Implements a search bias based on the idea the search space was a good choice
  - good solutions may be symmetric w.r.t. the centre of the search space
  - If the global optimum is near the centre of the search space, then the search is more efficient.

Related algorithms

- Gravitational Search Algorithm
- Central force optimisation
  - includes acceleration (similar to PSO with $\omega > 0$)
  - nontrivial powers (like α, β in ACO)
- Charged System search

Hysteretic Optimisation (Ising 1925, Lin 2013)

- Encode a binary combinatorial optimisation problem by an Ising model

$$H(\{\sigma_i\}) = -\frac{1}{2}\sum_{ij}^{N} J_{ij}\sigma_i\sigma_j - \gamma\sum_i h_i\sigma_i$$

- $P(\{\sigma_i\}) = \frac{1}{Z}\exp(-\beta H(\{\sigma_i\}))$
- See SA how to change behaviour by changing β
- We can also change γ to couple the state $\{\sigma_i\}$ more tightly to the external field $h_i \in [-1, 1]$
- System undergoes a hysteresis when γ is changed in either direction.
- Applicable to all problems that can be encoded by J_{ij} and h_i.

Spiral-Based Search algorithms (Tamura & Yasuda, 2010)

- Assume x^* is the current best and a particle $x \neq x^*$ is supposed to spiral towards x^*
- Idea: Rotate the vector pointing from x^* towards x by a small angle φ and reduce the distance by a function $g < 1$, i.e.

$$x(t+1) = x^* + g(|x(t) - x^*|, t) R_{\varphi(t)} (x(t) - x^*)$$

- For $D > 2$ we need more than one angles, namely $\frac{1}{2} D(D-1)$, or choose randomly two of the dimension and apply the rotation $\begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix}$ to these two.

Spiral-Based Search algorithms (Tamura & Yasuda, 2010)

- Variants:
  - logarithmic, Archimedean, lituus (distance decays as $\varphi^{-1/2}$)
  - fitness-dependent outwards or inwards spirals ($g \gtrless 1$)
  - random angles, random decay, random noise
  - spirals alternatingly w.r.t. several bests
- Problem:
  Coverage of search space not good in higher dimensions
- Advantage: More directly controllable (no scales or norms)
- Search bias: Local

A generic algorithm

It seems that most of the continuous metaheuristic optimisation algorithms can be represented by the following scheme

- For population of vectors $x_i \in \mathbb{R}^d$

$$v_i(t+1) = \omega v_i(t) + \alpha \sum_{j,k} \rho \circ (\hat{x}_j(t) - \hat{x}_k(t))$$

$$x_i(t+1) = \begin{cases} x_i(t) + v_i(t+1) & \text{with Prob. } 1-p \\ \kappa x_i(t) + \xi & \text{with Prob. } p \end{cases}$$

- $\hat{x}_j$, $\hat{x}_k$ can be global best, group best or random other more or less fit individuals
- ρ can be 0 or random from an interval $[\rho_{\min}, \rho_{\max}]$, and can be different for different dimensions. It can incorporate a fitness-related probability. Often $\omega = 0$ and $\alpha = 1$
- ξ denotes the replacement of one individual by a random solution ($\kappa = 0$) or the addition of noise ($\kappa = 1$)

Metaphors in Metaheuristic Optimisation

- The novelty of the underlying metaphor does not automatically render the resulting framework "novel".
- There is increasing evidence that very few of the metaphor-based methods are new in any interesting sense.

From Sörensen (2013) following Scholarpedia “metaheuristics”

How does this relate to the NFL theorem?

- Exploration vs. exploitation: Assume that nearby solutions are of similar quality
- Information about the search space: Minimal prior knowledge is always available
- Extraction of information on the problem: Assuming that initial data have similar properties as later data or that they allow making inferences about later data
- Pragmatic evaluation: It is the performance on your problem that counts (.. also include cost for time to model, program, explain to any users etc.)

Preliminary Conclusion: What questions to be asked?

- How can we control the balance of exploration vs. exploitation?
- How to include known information about the search space?
  - Size, dimension, structure (see below)
- Does the algorithm extract and represent any kind of information on the problem
  - Step sizes, length scales, adaptive parameters, adaptive methodology
  - Dimensionality, constraints, sub-manifolds
  - Directionality, gradient-like information, correlations
  - Patterns, higher-order correlations, heterogeneity, compositionality, building blocks, self-similarity, ...
- How do we evaluate the algorithm for a problem of interest?
  - Better than random walk?
  - How many fitness evaluations?
  - How intuitive?

Conclusion & Outlook

- MHO Algorithms from physics are not necessarily "better" than algorithms from biology: Inspiration is not enough.
- Biologically-inspired algorithm are designed based on low dimensional intuition, physics-inspired likewise,
- Biologically optimisation is usually w.r.t. a niche, so physics-based algorithms may have more potential to generalise (with the exception of *neural networks*)
- What matters is the alignment of the algorithm: What can we find out about a problem?
  - Information theoretic analysis (Steer et al. 2008)
  - Exploratory landscape analysis (Mersmann et al., 2011)
  - Hyperheuristic algorithms
- How do we choose algorithms in a particular application?