Course: Natural Computing

*7. Metaheuristics in Contex

including also: Convergence and Remarks on biological evolution

J. Michael Herrmann

School of Informatics, University of Edinburgh

michael.herrmann@ed.ac.uk, +44 131 6 517177

Overview

- Algorithmic bias
- Fisher's fundamental theorem
- History
- Perspectives

Algorithmic bias*

Definition: Statistical bias

- Sample: $\{X_i\}_{i=1,\ldots,N}$
- Estimator: $T(\{X_i\}) = \hat{\theta}$ of any property of the sample (e.g. of the mean)
- Bias: $\left\langle \hat{\theta} \right\rangle - \theta^*$ by definition, i.e. eviation from the *true value* θ^* of this property averaged over samples

Biases in machine learning

- Data: $\{X_i\}_{i=1,\ldots,N}$
- Algorithm (e.g. classification) $X_i \to \{0, 1\}$
- Bias: Swapping of data properties that are *irrelevant* for the decision affects the decision reached by the algorithm

*) There are more general approaches to algorithmic biases, i.e. we adopt a simplification that seems to cover most problems that can occur with MHO algorithms

- "A basic insight of machine learning is that prior knowledge is a necessary requirement for successful learning" Shai Ben-David et al. (2011) Universal learning vs. no free lunch results. In: Philosophy and Machine Learning Workshop NIPS. 2011.
- "you can't do inference ... without making assumptions" David MacKay (2003) "Information theory, inference and learning algorithms"

- **Block-uniform distributions guarantee no-free lunches**
  - A prior over functions is **block-uniform** if any two functions that are connected by a function permutation have the same prior probability.
  - If two fitness functions return the same fitness values although possibly in a different order, we say they are connected by a **function permutation**.
- More specifically, block-uniform distributions capture exactly the scenarios where no free lunch results hold for any metric.
- However, when we are interested in no free lunch results with respect to particular metrics, and for limited numbers of samples, then free lunches are possible even under block-uniform distributions.

see English (2004)

Algorithmic bias

- By design (e.g. search direction)
- Implementations of algorithms not error-free (human factors)
- Benchmarks vs. real world (design doesn't end at benchmarks)
- Above-mentioned problems with block-uniform distributions (imbalanced classes, uncertainty differences) [often technically solvable]
- By convention incl. analytical convenience: Assumptions of independence, normality, identical distribution, model certainty
- Active learning ("filter bubble")
- No known procedure to exclude all biases (in non-trivial domains there is not enough data to guarantee a statistically-valid evaluation of the many possible swaps)
- Only known biasses can be checked by swapping or omitting of critical attributes

Algorithmic bias in MHO

Reducing bias

- Clear concept of algorithms: Acquisition of new sample based on all previous samples
- MOO helps to avoid commitment to questionable fitness functions
- MHO algorithms make few assumptions
- Algorithms are designed to avoid local optima

Generating bias

- Active learning: New samples are found based on previous samples (efficiency is irrelevant)
- SOO Fitness functions usually not questioned, user chooses termination and initialisation
- Design towards simple solutions (e.g. axis bias)
- Metaphors do not help to uncover biases

Conclusion: Algorithmic bias in MHO

- MHO carries the potential of algorithmic bias like every other algorithm.
- Avoidance of local optima is one way to reduce algorithmic bias.
- Diversity is difficult to assess: Co-diversity approaches are needed, i.e. the assessment of diversity needs to follow the diversity principle itself.
- Algorithms are quite flexible and are usually adapted and evaluated by the practitioners themselves, which reduces the reality gap.

Outlook

- Applications of MHO
- Natural computing and neural networks
- Population-based robotics and swarm intelligence
- Recent trends and open questions

Course: Natural Computing

*6. Convergence of MHO

J. Michael Herrmann

School of Informatics, University of Edinburgh

michael.herrmann@ed.ac.uk, +44 131 6 517177

Overview

- Convergence
- Parameters
- Evaluation of MHO
- Larmarck's evolution theory and the Baldwin effect

Reminder: Convergence in PSO

- **Failure**: Swarm diverges or is stopped by search space boundaries
- **Ideally**: Global best approaches global optimum while swarm converges
- **Typically**:
  - Global best approaches a local optimum because premature collapse of the swarm
  - Global best is near global optimum and swarm remains itinerant
- Convergence may be useful to search the space around a good solution more carefully
- Convergence is not necessary (global or local bests remember previous good solutions)
- A final local search stage can locate the precise positions of the nearest optimum

Two meanings of convergence

(A) Dynamics comes to a halt[1]

1. GA: All individuals in a population are identical
2. PSO: All particles converge to a single point and velocities approach zero
3. ACO: All ants take the same path

Many techniques available: E.g. Lyapunov stability.

(B) Global optimum is found

1. GA: one individual has maximal fitness
2. PSO: the absolute difference of the fitness of the best-so-far solution and the maximal fitness is smaller than an appropriate threshold
3. ACO: one ant has maximal fitness

Either need to know optimal fitness or need to prove that global optimum is found (e.g. with probability 1)

---

[1] Stochastic case: Dynamics becomes stationary, i.e. distribution of solutions converges

Convergence to global optimum: ACO

We ignore for the moment the dynamics question (A), and ask about the global optimum (B) for ACO (Stützle & Dorigo, 2002)

(B.1) The pheromone trails along the path representing the optimal solutions are larger than on any other solution

(B.2) Probability that an ant finds the globally optimal solution approaches 1 after sufficiently long time

In the following we will assume that only one global optimum exists, and will consider the Min-Max Ant System algorithm with best-ant pheromone update

1. Establishing a pheromone trail

Assume the best path S^* was found by an at time t^*. Let (i,j) be a component in S^*, but (worst case) $\tau_{ij}(t^*) = \tau_{\min}$ and all $(k,l) \notin S^*$ have $\tau_{kl} = \tau_{\max}$. If only the best ant lays pheromones, then the level on S^* will increase within t time steps by

$$\tau_{ij}(t^* + t) = \rho^t \tau_{\min} + \sum_{s=1}^{t} \rho^{s-1} \Delta\tau_{\max}$$
$$> t\,\rho^{t-1} \Delta\tau_{\max} = t\,\rho^{t-1}(1-\rho)\,\tau_{\max}$$

because $\tau_{\max}(t) = \rho^t \tau_{\text{init}} + \sum_{s=1}^{t} \rho^{t-s} \Delta\tau_{\max}$, i.e. $\tau_{\max} = \frac{\Delta\tau_{\max}}{1-\rho}$,

whereas (assuming $\tau_{\min} < \rho^t \tau_{\max}$) $\tau_{kl}(t^* + t) = \rho^t \tau_{\max}$ such that

$$\tau_{ij}(t^* + t) > \tau_{kl}(t^* + t) \quad \text{if} \quad t > \frac{\rho}{1-\rho}.$$

2. Finding a global optimum: ACO

Let $P^*(t)$ denote the probability that the best path is found at least once by time t. We need to show that (Stützle & Dorigo, 2002)

$$\forall \varepsilon > 0 \, \exists t : P^*(t) > 1 - \varepsilon.$$

We assume for simplicity that we have a single ant only, the exponent $\alpha = 1$, the local desirability is constant, and each step of the solution has at most K branches.

In the worst case the optimal path has a pheromone level $\tau_{\min} > 0$, the other $K - 1$ branches have $\tau_{\max}$. The probability rule gives:

$$p_{\min} = \frac{\tau_{\min}}{\tau_{\min} + (K-1)\tau_{\max}}$$

Therefore, we have $P^*(1) \geq p_{\min}^D > 0$, where D is the number of components in the solution. The proof is completed by noticing:

$$P^*(t) \geq 1 - \left(1 - p_{\min}^D\right)^t$$

Convergence to global optimum: Remarks

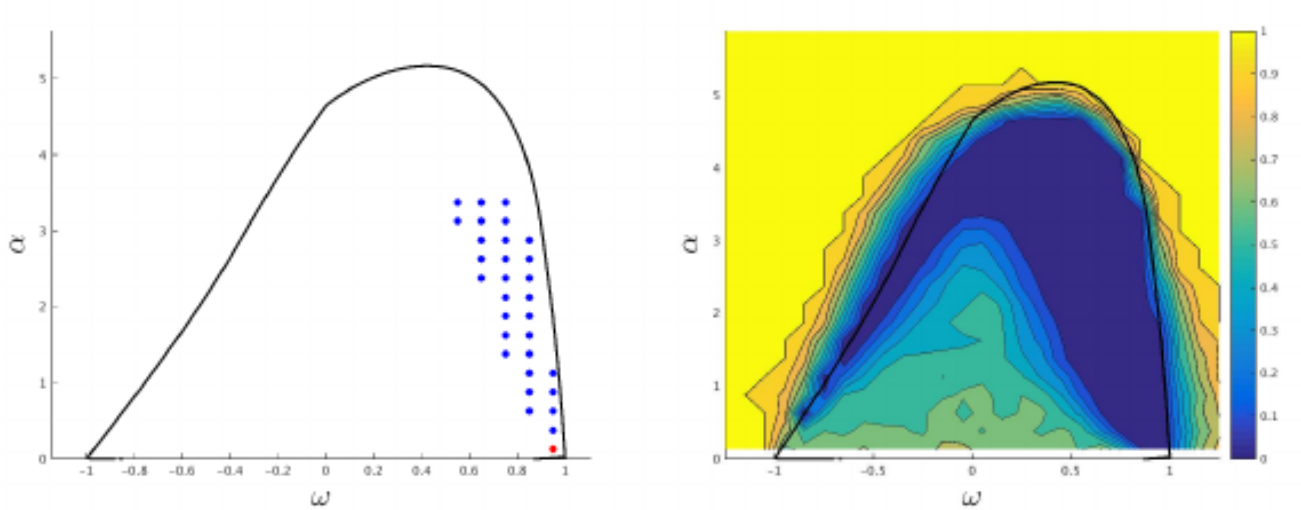
- $p_{\min}^{D}$ is very small, so the convergence time bound is very large
- A tighter bound is implied for more ants, by considering that there are fewer branches for decisions down the path, and that the pheromones are usually more fortunately distributed, but
- the bound is still exponential.
- Local desirability may reduce complexity, but may also impede convergence to global optimum, if the problem is deceptive
- The update by best ant only and the fact that $\tau_{\min} > 0$ are important (convergence questionable for other ant algorithms)
- See also: W. Gutjahr (2000) A graph-based Ant System and its convergence. Future Generation Computer Systems 16, 873–888.
- A similar proof can be given for GA with mutation rate $p_m > 0$

Convergence to global optimum for PSO?

- Statements similar to (B.2) were made by F. v. d. Bergh (2001) and Liu, Abraham & Snasel (2009)
- To show that the algorithm eventually finds the global optimum, we need to assert that a particle can get close with some (possibly very small) probability to every point in state space e.g. by
  - using Gaussian noise, i.e. the forces become $\zeta_1 (\mathrm{p} - \mathrm{x})$ with $\zeta_1 \sim \mathcal{N}\left(\frac{\alpha_1}{2}, \sigma^2\right)$ and $\zeta_2 (\mathrm{g} - \mathrm{x})$ with $\zeta_2 \sim \mathcal{N}\left(\frac{\alpha_2}{2}, \sigma^2\right)$
  - including a random walk to diversify the particle positions
  - choosing parameters such that particles perform independent random movements through all dimensions of the whole search space.
- The noise-based approaches may counteract exploitation, so an appropriate choice of the parameters seems preferable to reach a good level of exploratoriness

Implications from numerical experiments

PSO performance for the minimisation of a sphere function $\sum_i x_i^2$ for the relevant pairs (α, ω) with $\alpha = \alpha_1 + \alpha_2$ and $\alpha_1 = \alpha_2$.



- Numerical experiments show that the best results (dots in the left image) are obtained in a region similar to the oscillatory region indicated by the simplified model (see previous slides)
- In contrast to the simplified model we find
  - near $\omega = 1$, good performance is possible only for small α
  - good results are possible also for negative ω, see regions with small average deviations from global optimum (right image)
  - good results are possible also for $\alpha > 4$ for moderate ω
- For $\alpha_1 \neq \alpha_2$, the curve is similar but not identical.

Conclusion on Convergence

- It is relatively easy to show that MHO algorithms can find the global optimum of an arbitrary search problem, but these proofs are not practically useful as they imply an exponentially long run time
- It is more important to find parameter settings that help to speed up the search,
  - a few rules exist how to choose parameter values in general (except for PSO and DE)
  - for a specific problem, practical experiences are needed in order to find optimal parameters (later material on applications)
  - sometimes a higher-order MHO algorithms is employed for the parameter search (later material on hyperheuristic algorithms)

Mean or best result on a number of runs?

- *mean* $\pm$ *standard deviation* is the representation of choice
- *mean* $-$ *standard deviation* can be negative even for a positive random variable: use one-sided std. dev.
- best result over a number of runs is
  - overly optimistic
  - probably not robust
  - may be an outlier
- *variance* or even *mean* may not exist (e.g. for certain probability distributions) or does not make sense, e.g., if for some random initialisations the algorithm diverges while it performs well for others). Often the median can be used instead.
- Best result can be useful in applications, if sufficiently robust

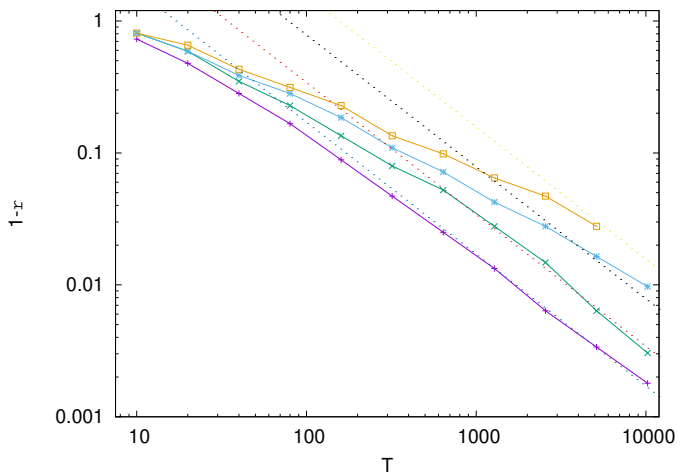
Birattari & Dorigo (2007) How to assess and report the performance of a stochastic algorithm on a benchmark problem: *mean* or *best* result on a number of runs? *Optimization Letters* **1**:309–311.

Scaling analysis

Distinguish between different forms of scaling

- Scaling of performance with problems size or dimensionality (complexity)
- Scaling w.r.t. to termination criterion (precision)
- Scaling of population (populations often quite small)

Warning: MHO algorithms sometime scale irregularly, i.e. they may scale well for medium problem sizes, but are exponential at larger sizes (for exponential or NP-complete problems)



Example: Optimal parameter $1-\rho$ as function of runtime T for TSPs with 50, 100, 200 and 400 cities. The relation is nearly a powerlaw, i.e. ρ approaches 1 for large T, but more slowly for larger problems.

Lessons from Natural Evolution

Course: Natural Computing (week 6)

Lamarckian evolution theory and the Baldwin effect

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

GA: Behaviour near the optimal solution

[De Jong] Say range of fitness values is [1,100]. Quickly get population with fitness say in [99,100]. Selective differential between best individual and rest, e.g. 99.988 and 100 is very small.

GA tends not to prefer one over the other: Balance between selection-induced improvement and mutation induced reduction of fitness. What can be done?

- Dynamically scale fitness as a function of generations or fitness range
- Use rank-proportional selection to main a constant selection differential. Slows down initial convergence but increases "exploitation" in the final stages.
- Elitism. Keep best individual so far, or, selectively replace worst members of population
- Shift balance from exploration at start to exploitation towards the end of the allocated time budget: Parameter control
- Compact GA (CGA) [Harik e.a., 1999]: A largely theoretical approach, which can be used to identify which genes are decided and which are still fluctuating in the population (prone to local optima).

Lamarckism

Characterised by

- Inheritance of acquired traits
- Use and disuse determine characteristics

More specfically, Lamarck provided a systematic theoretical framework for understanding evolution as the interplay of two processes

- **A complexifying force:** in which the natural, alchemical movements of fluids would etch out organs from tissues, leading to ever more complex construction regardless of the organ's use or disuse. This would drive organisms from simple to complex forms.
- **An adaptive force:** in which the use and disuse of characters led organisms to become more adapted to their environment. This would take organisms sideways off the path from simple to complex, specialising them for their environment.

wikipedia on Lamarck

The Baldwin effect

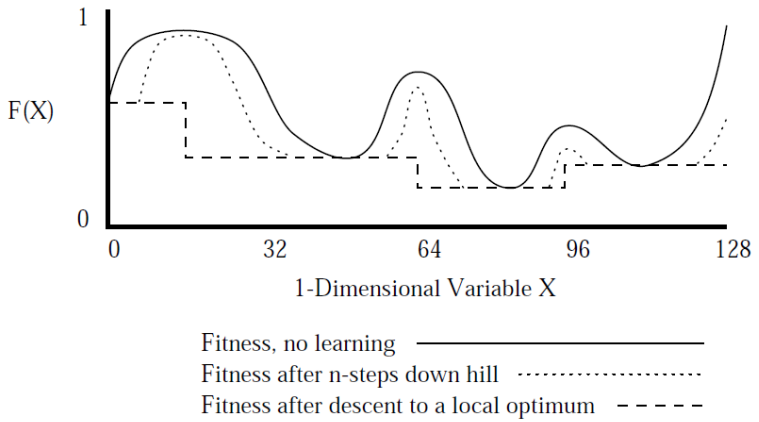
- "A new factor in evolution" (James Baldwin, 1896)
- Selection for learning ability (rather than relying only on fixed abilities from the genes)
- Increased flexibility: Robustness to changes in the environment (i.e. changes of the fitness function)


[University of Toronto]

- Selective pressure may lead to a translation of learned abilities into genetic information!
  - Learning has a cost
  - If learning of the same tasks increases fitness over many generations then those individuals have a relatively higher fitness that produce (parts of) these results by their genetically fixed abilities

Computational studies

- Hinton & Nowlan: How learning can guide evolution (1987) (see M. Mitchell, Chapter 3)
  - Binary genome plus undecided bits which are set in "life" by learning
- Whitley, Gordon & Mathias: Lamarckian evolution, the Baldwin effect and function optimisation (1994)
- Standard GA (elitist) plus: Lamarckian evolution (editing strings) or Baldwinian evolution (adaptation process before fitness determination)



[technically in both cases: hill-climbing in the fitness landscape]

- Lamarckian faster, but Baldwinian less likely to be trapped by local optima

Conclusion

- It is relatively easy to show that MHO algorithms can find the global optimum of an arbitrary search problem, but these proofs are not practically useful as the imply an exponentially long run time
- It is more important to find parameter settings that help to speed up the search,
  - For some algorithms, rules exist how to choose parameter values in general
  - for a specific problem, practical experiences are needed in order to find optimal parameters
  - sometimes a higher-order MHO algorithms is employed for the parameter search
- Adding a heuristic to a MH achieve a final improvement is also often advisable as a form of postprocessing