Course: Natural Computing

10. Recent Advances in Metaheuristic Optimisation

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

Overview

- 10 successes of MHO
- Other current trends in MHO
- Special problems
- 10 challenges for MHO

History of MHO (according to Wikipedia)

1952: H. Robbins and S. Monro work on stochastic optimisation methods.
1954: N. A. Barricelli: Evolution for general optimisation problems.
1963: L. A. Rastrigin proposes random search.
1965: J. Matyas proposes random optimisation.
1965: J. A. Nelder and R. Mead propose a simplex heuristic.
1965: Ingo Rechenberg discovers the first Evolution Strategies algorithm.
1966: Lawrence J. Fogel et al. propose evolutionary programming.
1970: W. K. Hastings proposes the Metropolis–Hastings algorithm.
1970: D. J. Cavicchio: adaptation of control parameters for an optimizer.
1970: W.B. Kernighan, S. Lin: Graph partitioning method (tabu search).
1975: John H. Holland proposes the genetic algorithm.
1977: Fred W. Glover proposes scatter search for integer programming.
1978: R. E. Mercer, J. R. Sampson: Metaplan for parameter tuning.
1980: Stephen F. Smith describes genetic programming.
1983: S. Kirkpatrick et al. propose simulated annealing.
1986: Fred W. Glover: Tabu search, first mention of *metaheuristic*.
1989: P. Moscato proposes memetic algorithms.
1990: G. Dueck, T. Scheuer: Threshold accepting (deterministic SA).
1995: D. H. Wolpert and W. G. Macready prove the no free lunch theorems.

10 Successes of MHO*

1. 1990: Threshold accepting (Deterministic simulated annealing)
2. 1995: XCS: Classifier fitness based on accuracy.
3. 1996: John R. Koza's electronic circuits
4. 2000: The GOLEM project
5. 2008: Digital Image Evolution of Artwork: Mona Lisa problem
6. 2017: Differentiable GP
7. 2018: Playing Atari games with GP
8. 2020: AI Feynman
9. 2022: Evolution through Large Models
10. 2022: Intelligence as Capacity for Exploration

*Choice is largely subjective. Any additions?

1. Threshold accepting

- Simulated annealing without stochasticity[1]
- Optimisation w.r.t. a set of nearest neighbours (for each city in TSP)
- Accept if quality Q of new state is not below Q-T for some T
- Instead of varying temperature: Adapt T, e.g. decrease T regularly (when stuck, increase T, or restart)
- Algorithm is more data-driven and less chance-driven, but it helps if best fitness is known
- Proven to be better than SA (and similar algorithms)[2]

[1] G. Dueck, and T. Scheuer, 1990. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics* **90**(1), 161-175.

[2] A. Franz, K. H. Hoffmann, P. Salamon, 2001. Best possible strategy for finding ground states. *Physical Review Letters* **86**, 5219.

2. eXtended Classifier Systems (XCS)

- Classifier systems: GA can help to select and shape classifiers
- GA evolves schemata that are used to match inputs to the system
- A low-predicting classifier may nevertheless be the best one for its environmental niche.
- XCS fitness: Accuracy of prediction, instead of prediction itself.
- Action selection updated by Q-learning
- Rule-based machine learning. Combining GA and RL.

Wilson, S.W., 1995. Classifier fitness based on accuracy. Evolutionary Computation **3**(2), 149-175.

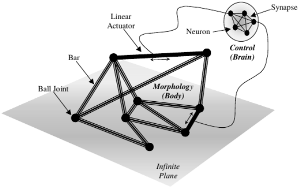
Layout and sizing (numerical values) of electronic circuits:

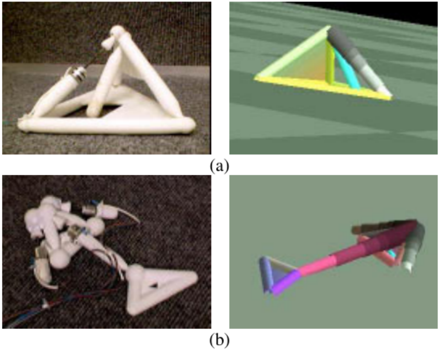
- crossover (woofer and tweeter) filter
- low-pass filter
- amplifier
- asymmetric bandpass filter

Koza, J.R., Bennett, F.H., Andre, D. and Keane, M.A., 1996, May. Four problems for which a computer program evolved by genetic programming is competitive with human performance. In Proceedings of IEEE International Conference on Evolutionary Computation (pp. 1-10). IEEE.

4. “Genetically Organized Lifelike Electro Mechanics”

Virtual diversity of morphology + control in reality by 3D solid printing





Pollack, J.B. and Lipson, H., 2000, July. The GOLEM project: Evolving hardware bodies and brains. In Proceedings. The Second NASA/DoD Workshop on Evolvable Hardware (pp. 37-42). IEEE.

⇒ Evolutionary robotics


Original Image


Generation 12,974


Generation 904,314

- Genetic algorithm chooses 50 semitransparent polygons
- Published online by Roger Alsing (?) and Roger Johansson, 2008. https://rogerjohansson.blog/2008/12/07/genetic-programming-evolution-of-mona-lisa/
- Garbaruk, Julia, et al. (2022) Digital Image Evolution of Artwork Without Human Evaluation Using the Example of the Evolving Mona Lisa Problem. *Vietnam Journal of Computer Science* **9**.02, 203-215.
- Try it yourself at alteredqualia.com

6. Differentiable genetic programming

dCGP: Cartesian GP with automatic differentiation

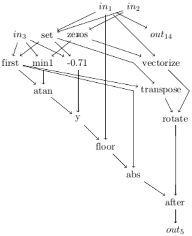
- Restrict alphabet to differentiable functions (may include approximations, e.g. for division)
- Gradient based adaption of constants or parameters
- Explicit derivatives to find analytic solutions of differential equations

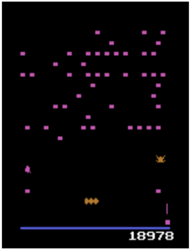
Izzo, D., Biscani, F. and Mereta, A. (2017) Differentiable genetic programming. In European Conference on Genetic Programming, 35-51.

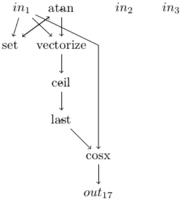
7. Atari games by Genetic programming

V. Mnih et al. (2013) Playing Atari with deep reinforcement learning. arXiv:1312.5602.

D. G. Wilson et al.: Evolving simple programs for playing Atari games (GECCO'18)1







"*Neural networks have garnered all the headlines, but a much more powerful approach is waiting in the wings.*"

MIT Technology Review (18/7/18) Cartesian genetic programming outperforms deep-learning machines at Playing Atari Games.

8. AI Feynman

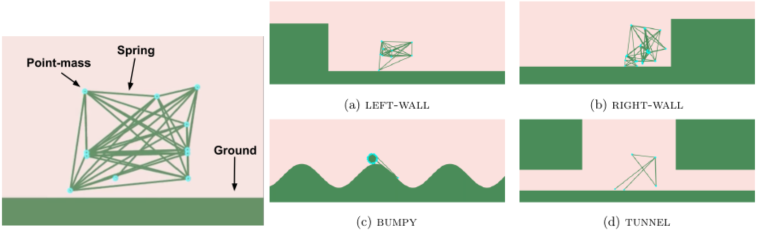
- Symbolic regression matches data to understand the target domain
- Neural network fitting with a suite of physics-inspired techniques.
- Target: 100 equations from the Feynman Lectures on Physics (discovers all of them)
- Equations contain what is understood about physics, so project carries potential for transparency, explainability, interpretability
- Fitness should also include: symmetries, separability, compositionality, understandability

Udrescu, S.M. and Tegmark, M. (2020) AI Feynman: A physics-inspired method for symbolic regression. *Science Advances* **6**:16, p. eaay2631.

9. Large models at OpenAI

Evolution through large models (ELM)

- Represent diff mutations by a large language model (300M parameters)
- Diff mutations clearly outperform trivial GP operators
- Fine-tuning by RL (walking machines on various terrains)

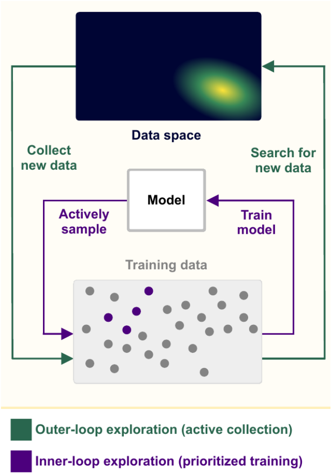


Lehman, J., Gordon, J., Jain, S., Ndousse, K., Yeh, C. and Stanley, K.O., 2022. Evolution through Large Models. arXiv preprint arXiv:2206.08896.

10. Artificial General Intelligence

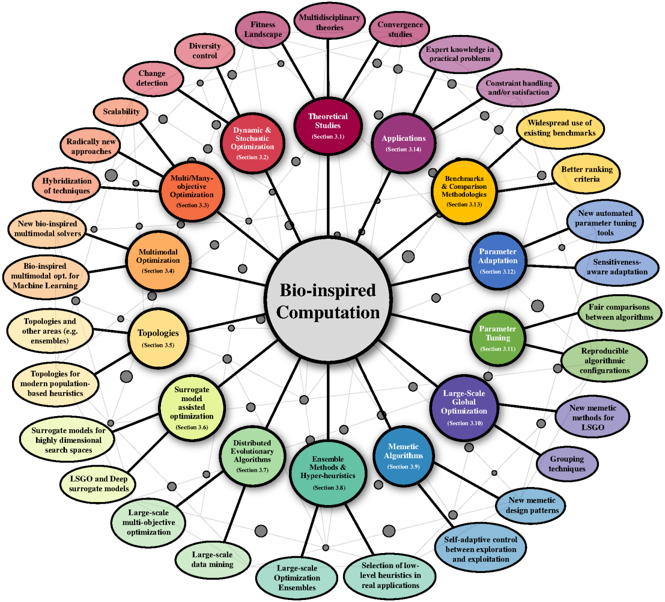
"Intelligence as Capacity for Exploration"

1. Improvability: An open world is never fully explored
2. Learnability: Open-ended learning includes structural learning
3. Consistency across environments implies semantics (?)



Jiang, M., Rocktäschel, T. and Grefenstette, E., 2022. General Intelligence Requires Rethinking Exploration. arXiv preprint arXiv:2211.07819.

Recent problems in MHO



J. Del Sera et al. (2019) Bio-inspired computation: Where we stand and what's next.

More interesting topics

- Explainability
- Co-evolution
- Ensembles (DE)
- Pareto
- Graphs
- Machine learning
- Set-oriented numerical optimisation [9]
- More papers [6][2][3]

Genetic programming for explainable AI

- Fitness is often costly when evaluated in the real world
- If the fitness is obtained by a computational model then
  - the model may not be good
  - optimisation may be possible in an easier way
- However, if the model is good and complex and needs to be simplified then we have an ideal case for applying a metaheuristic approach, as follows
  - train a deep neural network (NN)
  - use the network to calculate the fitness of a GP that is supposed to represent the function of the network in few simple and understandable steps
  - If the GP is perfect it may replace the NN
  - If the GP does not represent the NN perfectly, it may still provide the explanatory power that is missing in the black-box approach with high representational power
  - It is not clear whether the GP may also indicate shortcomings of the NN

Co-evolution

- Generic example: Hardware and software
- Co-evolving parasites facilitate escape from local optima (W. D. Hillis, Physica D, 1990)
- Gene-culture co-evolution: Baldwin effect, memetic algorithms
- Evolving co-operative agents (morphing, clustering, composition, ...)
- In MHO algorithms as an attempt to reach larger problem sizes
  - Grouping of variable and decomposition of problems
  - Subproblem resource allocation
  - Co-operator selection based on mutual fitness
  - Fitness shaping and MOO
  - Representation learning

X. Ma et al. (2019) A Survey on Cooperative Co-Evolutionary Algorithms

Ensemble of Differential Evolution Variants (EDEV)

- Multi-population based framework (MPF) consisting of 3 DE variants (sizes, e.g., 10 + 10 + 10)
  - JADE (adaptive differential evolution with optional external archive)
  - CoDE (differential evolution with composite trial vector generation strategies and control parameters)
  - EPSDE (differential evolution algorithm with ensemble of parameters and mutation strategies)
- And a fourth (larger) population is used to produce results informed by the three indicator DEs (size, e.g., 70)

Guohua Wu, Xin Shen, Haifeng Li, Huangke Chen, Anping Lin, P. N. Suganthan
Ensemble of differential evolution variants (2018)

Pareto or Non-Pareto

- Problems with Pareto-style optimisation
  - slow convergence
  - poor representation of the front for many objectives
- The Pareto subpopulation provides diversity, the non-Pareto subpopulation exploitation
- Applicable to single-objective problems
- See also: Quality diversity [5] and multicriterial search [1]

Miqing Li, Shengxiang Yang, Xiaohui Liu (2016)
Pareto or Non-Pareto: Bi-Criterion Evolution in MOO

Combinatorial Optimisation Algorithms over Graphs

Problem Statement: Given a graph optimisation problem G and a distribution D of problem instances, can we learn a heuristics that generalize to unseen instances from D?

- Learning heuristic algorithms that exploit the structure of recurring problems
- Combination of reinforcement learning and graph embedding
- Greedy policy incrementally constructs solutions
- Framework can be applied to a range of optimisation problems: Minimum Vertex Cover, Maximum Cut, and TSP
- Good, but suboptimal results for up to 100 nodes

Hanjun Dai, Elias B. Khalil, Yuyu Zhang, Bistra Dilkina, Le Song (2017)
Learning Combinatorial Optimization Algorithms over Graphs

Machine learning and MHO

- For machine learning (optimise existing approaches)
  - Hyperparameter selection
  - Structure learning (e.g. network layout)
  - Design of exploration strategies (including restarts)
  - Identification of counterexamples
  - Ensemble methods

- From machine learning (map problems to algorithms)
  - Classification of problems in terms of algorithm applicability
  - Performance prediction for given instances (e.g. by SVM)
  - Learning of design properties of algorithms
  - Representation of information acquired about the problem
    (e.g. model for fitness function, search direction, Pareto fronts)

See also review by Heda Song, Isaac Triguero, and Ender Özcan, 2019 [7]

Recent problems in MHO: Neuroevolution

- Sigmoid, ReLU or Gaussian? Why not try to evove an activation function?

Yuen, B., Hoang, M.T., Dong, X. and Lu, T., 2021. Universal activation function for machine learning. Scientific reports, 11(1), p.18757.

Recent problems in MHO: Learning from biology

Dimension	In biology	In computation	Example computational approaches	Computational challenge	Opportunity
Openendedness	Apparent	Incipient	Avida, GenProg, Division Blocks[21,35,37]	Nonstationary fitness	Scale
Major transitions	Transformative	Minor	EVC, SEAM, Model-S[41–43]	Encapsulation	Develop biology
Neutrality and drift	Essential	Minimal	Novelty Search, MGA, GEVO[54,55,60]	Selection pressure	Emulate biology
Multi-objectivity	Implicit	Explicit	NSGA, Lexicase, Quality Diversity [72,73,78]	Dimensionality	Scale
Geno/pheno mappings	High DoF	Low DoF	CE, HyperNEAT, Deep GA[87–89]	Effective representations	Emulate biology
Co-evolution	Pervasive	Constrained	CoDeepNEAT, EUREQA, POET[23,102,103]	Generality	Scale

Miikkulainen, R. and Forrest, S., 2021. A biological perspective on evolutionary computation. Nature Machine Intelligence, 3(1), pp.9-15.

Recent problems in MHO: Theory

- We show analytically that training a neural network by conditioned stochastic mutation or neuroevolution of its weights is equivalent, in the limit of small mutations, to gradient descent on the loss function in the presence of Gaussian white noise.
- Averaged over independent realizations of the learning process, neuroevolution is equivalent to gradient descent on the loss function.

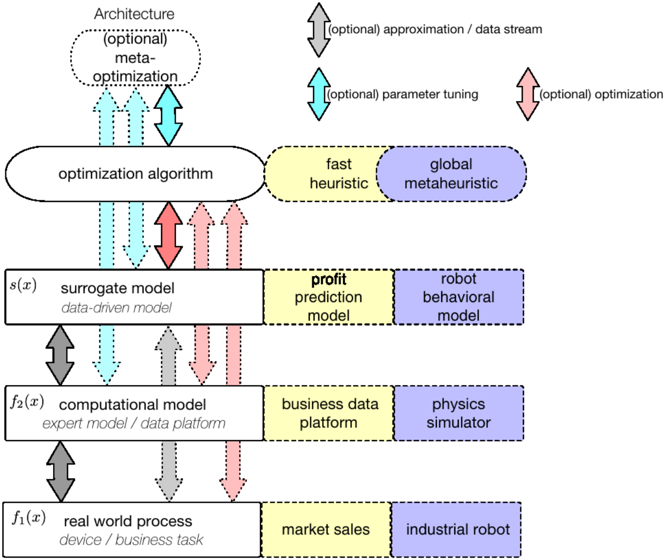
Whitelam, S., Selin, V., Park, S.W. and Tamblyn, I., 2021. Correspondence between neuroevolution and gradient descent. Nature communications, 12(1), p.6317.

Recent problems in MHO: Cleaning up

- Most MHO algorithms are variants of PSO, and some of these variants are not needed.

Camacho-Villalón, C.L., Dorigo, M. and Stützle, T., 2023. Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: six misleading optimization techniques inspired by bestial metaphors. International Transactions in Operational Research, 30(6), pp.2945-2971.

Machine learning and optimisation at various levels



Stork e.a.: A new taxonomy of global optimization algorithms. *Natural Computing* (2020) 1-24.[8]

MHO for computing beyond Moore's law

- Problems with classical computing
  - Physical limits and high energy consumption
  - Specs (MFLOPS) are rather are not really indicative for success
  - technical approaches may not be efficient for human-centred AI
- Bio-inspiration to be taken more seriously
  - Neural information processing at 0.1% energy consuption of present technical methods
  - Not all "bio" is neural (nor swarm-like): Analogue computing, meta-materials, morphological intelligence, active learning (Brooks: "Elephants don't play chess")
  - Evolution towards relevant computations

Ferdinand Peper (2017) The end of Moore's law: Opportunities for natural computing? *New Generation Computing* **35**:3, 253--269.

10 (interdependent) Challenges for MHO

1. Efficient hyperheuristics (or at least parameter tuning)
2. Alignment of problems and algorithms (including characterisation of landscapes and algorithms)
3. Criticality as a balance between exploration and exploitation [4]
4. Scalability (approaching hill-climbing for larger problems?)
5. Data-based search: Integration of neural networks (which can be useful also in other respects)
6. Dynamic and stochastic problems: mat-heuristics, learn-heuristics, sim-heuristics
7. Diversity and co-diversity: Diverse measures of diversity
8. Coevolution of algorithm (operators), population topology, search-space partition (building blocks), and solutions
9. Mathematical methods: Convergence, complexity, verifiation,...
10. Change from performance to scientific understanding (Kenneth Sörensen, Marc Sevaux, Fred Glover, 2017)

Specific problems to explore

- GP: Competition with neural network
- PSO: Evolutionary design of heterogeneous swarms
- IGA: Evolution of building blocks
- Reinforcement learning and MHO
- Robot swarms = Robots + MHO
- GP in HCI/HRI
- Check projects in Jan. 2024.

Conclusion on MHO

- MHO used to be a niche for developing simple algorithms
- It's not about metaphors, but about simplicity, experiments, and finding usable solutions quickly
- The many MHO algorithms are not very different from each other, but they include prior knowledge in different ways
- MHO algorithms gain their strength from teaming up with one another and with methods from outside MHO
- Perspectives are not too bad

Richard Allmendinger, Michael TM Emmerich, Jussi Hakanen, Yaochu Jin, and Enrico Rigoni.
Surrogate-assisted multicriteria optimization.
Journal of Multi-Criteria Decision Analysis, 24(1-2):5–24, 2017.

Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima.
Reference point specification in hypervolume calculation for fair comparison and efficient search.
In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 585–592, 2017.

Longmei Li, I. Yevseyeva, V. Basto-Fernandes, H. Trautmann, N. Jing, and M. Emmerich.
Building and using an ontology of preference-based multiobjective evolutionary algorithms.
In *Int. Conf. Evolutionary Multi-criterion Optimization*, pages 406–421. Springer, 2017.

Bernardo Morales-Castañeda, Daniel Zaldivar, Erik Cuevas, F. Fausto, and Alma Rodríguez.
A better balance in metaheuristic algorithms: Does it exist?
Swarm and Evolutionary Computation, 54:100671, 2020.

Justin K Pugh, Lisa B Soros, and Kenneth O Stanley.
Quality diversity: A new frontier for evolutionary computation.
Frontiers in Robotics and AI, 3:40, 2016.

Susanne Rosenthal and Markus Borschbach.
Design perspectives of an evolutionary process for multi-objective molecular optimization.
In *Int. Conf. Evolutionary Multi-Criterion Optimization*, pages 529–544. Springer, 2017.

Heda Song, Isaac Triguero, and Ender Özcan.
A review on the self and dual interactions between machine learning and optimisation.
Progress in Artificial Intelligence, 8(2):143–165, 2019.

Jörg Stork, Agoston E Eiben, and Thomas Bartz-Beielstein.
A new taxonomy of global optimization algorithms.
Natural Computing, pages 1–24, 2020.

Hao Wang, André Deutz, Thomas Bäck, and Michael Emmerich.
Hypervolume indicator gradient ascent multi-objective optimization.
In *Int. Conf Evolutionary Multi-Criterion Optimization*, pages 654–669. Springer, 2017.