Evolutionary Algorithms: An Introduction

Ahmed Hallawa
Anil Yaman
Dr. Giovanni Iacca

This project is funded by the European Union

Outline

Introduction

- Evolutionary Biology is a science concerned with
  - Diversity of life, differences and similarities among organisms
  - Adaptive and non-adaptive characteristics of organisms [3]

- Evolutionary algorithms (EA) is an umbrella term used to describe *Computer-based problem solving systems* which use computational models of evolutionary processes as key elements in their design and implementation [5]

Introduction

However, they all share the following attributes [15]:

- Individual: a candidate solution to a given problem
- Genotype: the genetic presentation of an individual
- Phenotype: the manifestation of genotype in an individual
- Fitness Functions: one or more function that associates a numerical score to each phenotype
- Population: the pool of multiple individuals undergoing evolution
- Selection: an operator that selects which individuals shall reproduce, based on their fitness
- Reproduction: one or more genetic operators (e.g. crossover & mutation that create new individuals from selected parents)
- Hereditary: parent and offspring present similar characteristics
- Diversity: individuals of the population are different to some extent
- Stopping criteria: one or more criterion used to stop the evolutionary process

Single-Objective EA

Algorithm 1 : Evolutionary Algorithm( )

```
procedure Standard_EA
    initialize P                              ▷ Population
    while not stopping criterion do
        F ← evaluate (P)                      ▷ Fitness
        P ← select (P, F)
        P ← reproduce (P, F)
    end while
end procedure
```

Algorithm	Citation
Evolutionary Strategy (ES)	Schwefel, 1966
Genetic Algorithms (GA)	Holland, 1988
Differential Evolutionary (DE)	Storn, 1997
Classifier Systems (CS)	Holland, 1977

Table: Single-objective EA

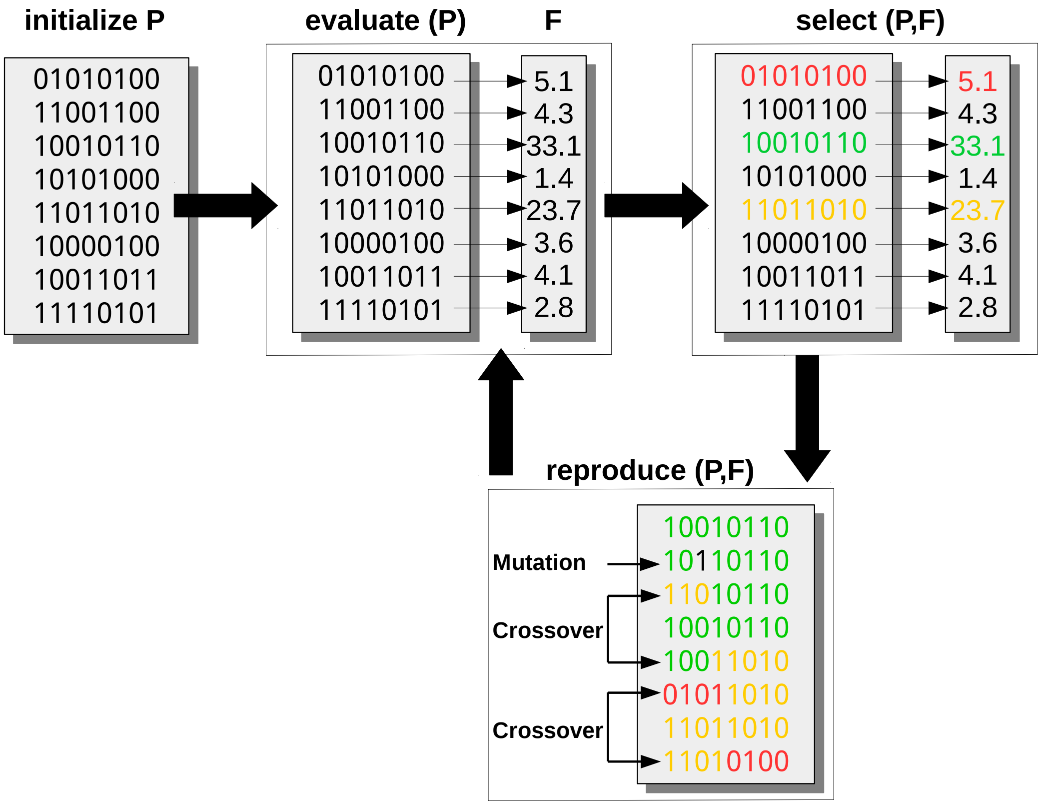


Figure: Evolutionary algorithm: an example

Multi-Objective EA

- The non-dominated set of all feasible solutions is called Pareto-optimal front
- Objective of multi-optimization algorithms is to converge to the Pareto-optimal front
- EA suit well multi-objective optimization problems due to their *population approach*

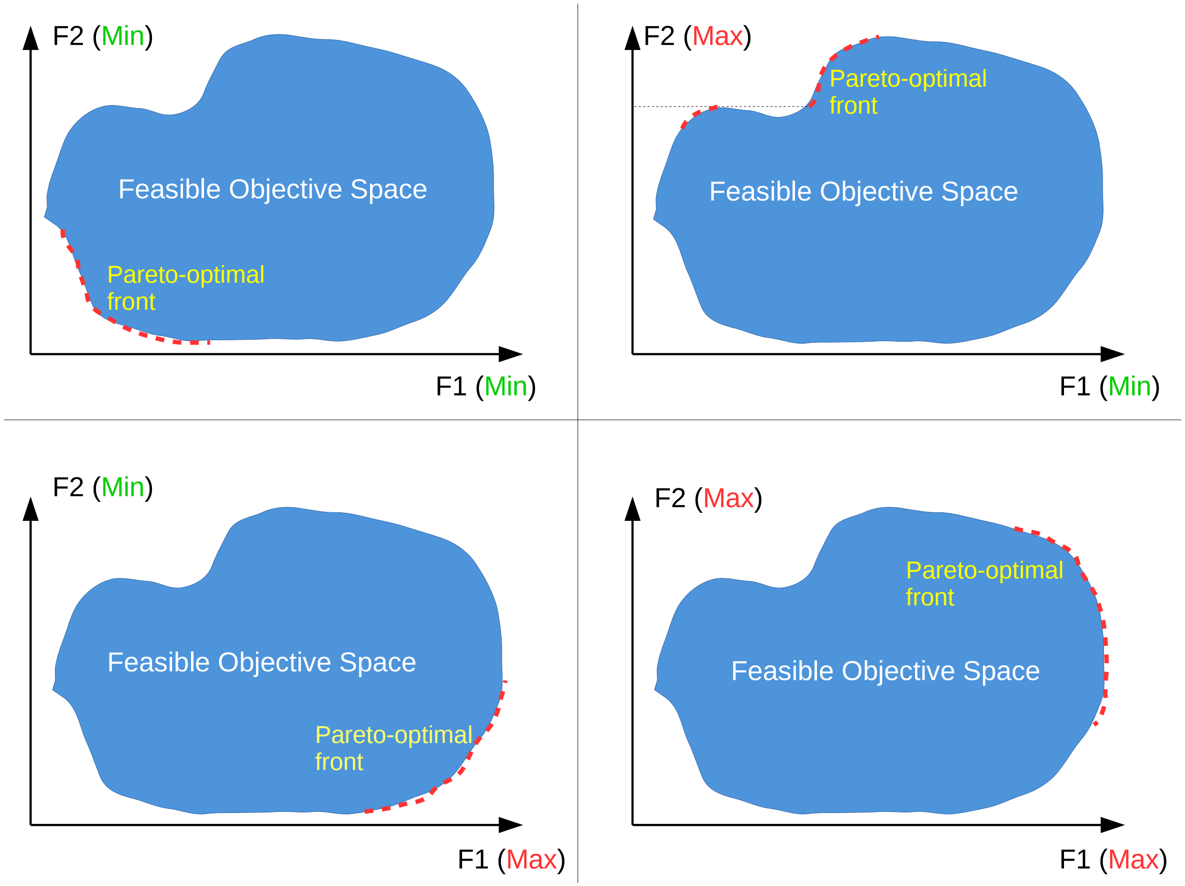


Figure: Pareto-optimal front: Examples

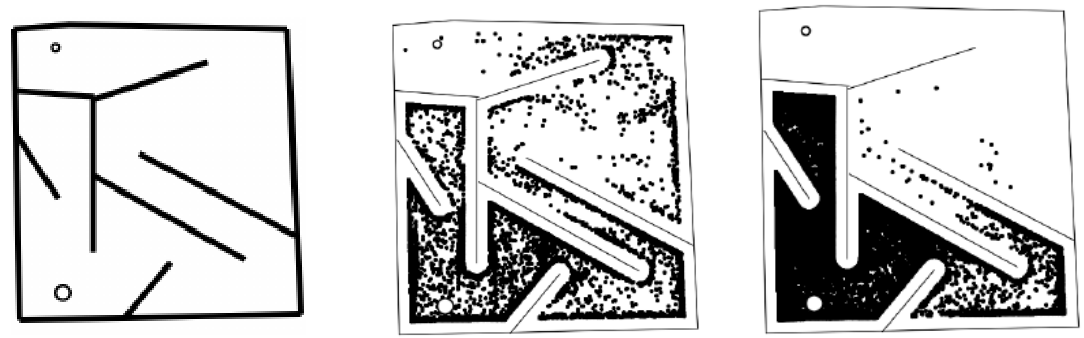
Multi-Objective EA

Algorithm	Citation
Vector evaluated GA (VEGA)	Schaffer, 1984
Vector optimized EA (VOEA)	Kursawe, 1990
Weight based GA (WBGA)	Hajela and Lin, 1993
Multiple objective GA (MOGA)	Fonseca and Fleming, 1993
Non-dominated sorting GA (NSGA)	Srinivas and Deb, 1994
Niched Pareto GA (NPGA)	Horn et al., 1994
Predator-prey ES	Laumanns et al., 1998

Table: Multi-objective evolutionary algorithms [4]

Objective-free EA

- An alternative idea is to abandon the goal of improving performance
- One subgroup of objective-free algorithms is called illuminating algorithms as they are designed to return the highest-performing solution in the feature space, thus illuminating the fitness potential of each region of that space



(a) Maze definition (b) NS (c) Fitness based

Figure: Novelty search: a comparison [9]

- Novelty search (NS) algorithm abandons the search for objective and rather searches for behavioural novelty. In many problems, it has shown supremacy against fitness based algorithms

Objective-Free EA

- MAP-Elites algorithm, on the other hand, attempts to simplify the search space with user defined features [13]

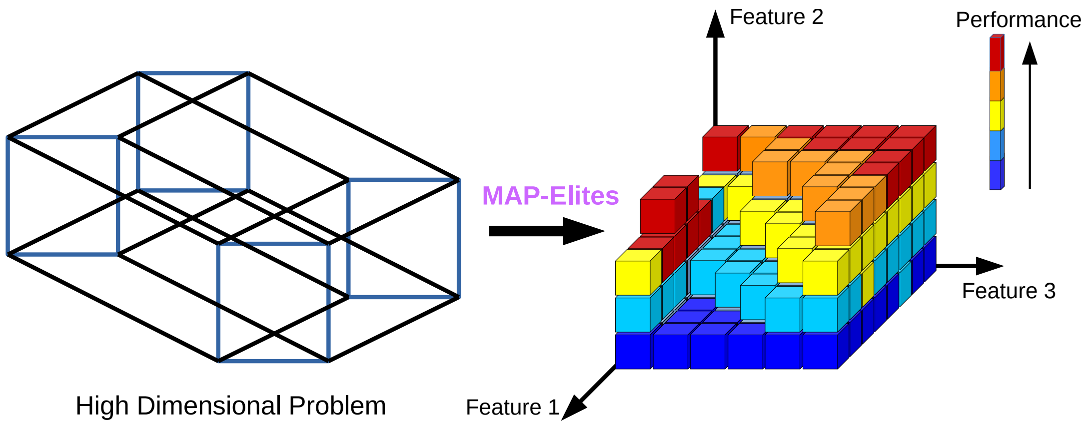


Figure: MAP-Elites algorithm

- MAP-Elites exploits parallelisms since the search for a solution in any *single cell* is aided by the simultaneous search for solutions in other cells

- MAP-Elites shows relationships between dimensions of interest and performance by illuminating fitness potential of the entire feature space, not only high-performing areas

Co-Evolution

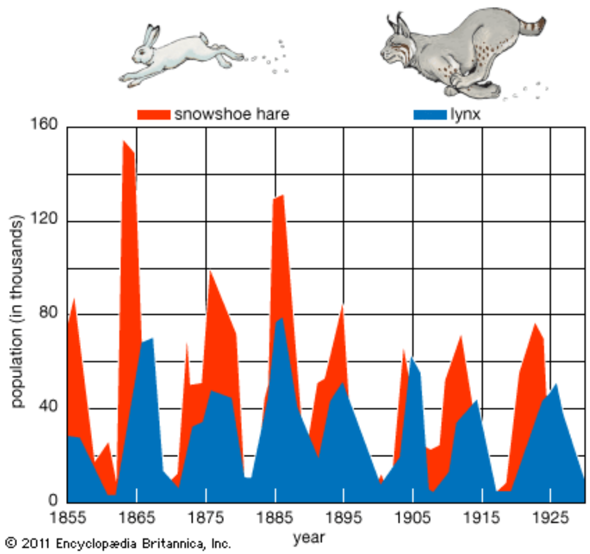


Figure: Predator-Prey Co-evolution

- Competitive co-evolution is where two different species co-evolve against each other e.g. predator-prey & parasite-host

- Helps prevent stagnation in local minima as it continuously change fitness landscape

- Increases adaptivity by producing an evolutionary arms race

- Minimizes the role of human-designed fitness function, thus more autonomous

- Problems such as *strategy recycling*, *dynamic fitness landscape* and *red queen effect* start emerging

Co-evolution

- Cooperative co-evolution is found in nature in many species
- Some cooperation schemes can't be easily modelled such as altruism
- Using adequate fitness function definition, co-operation can be realized (e.g. more fitness points granted for cooperation)
- Like competitive co-evolution, similar problems arise

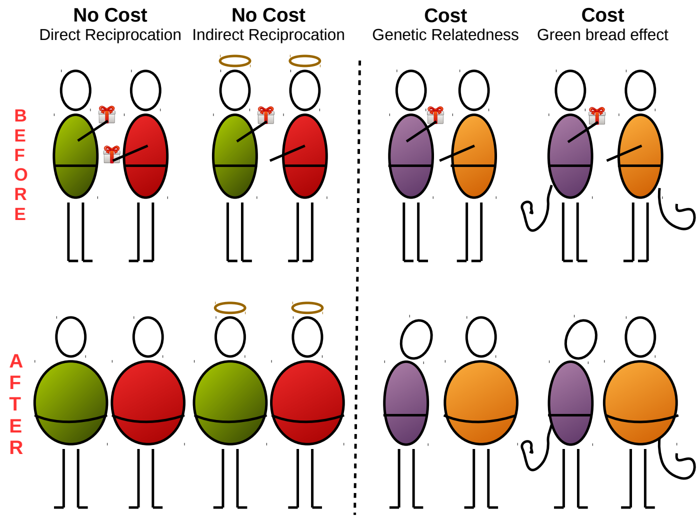
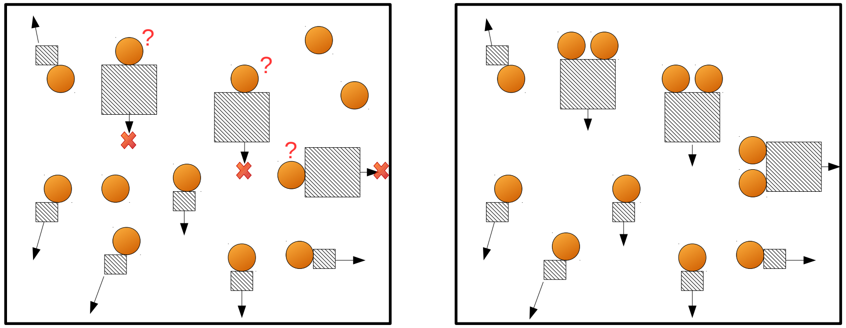


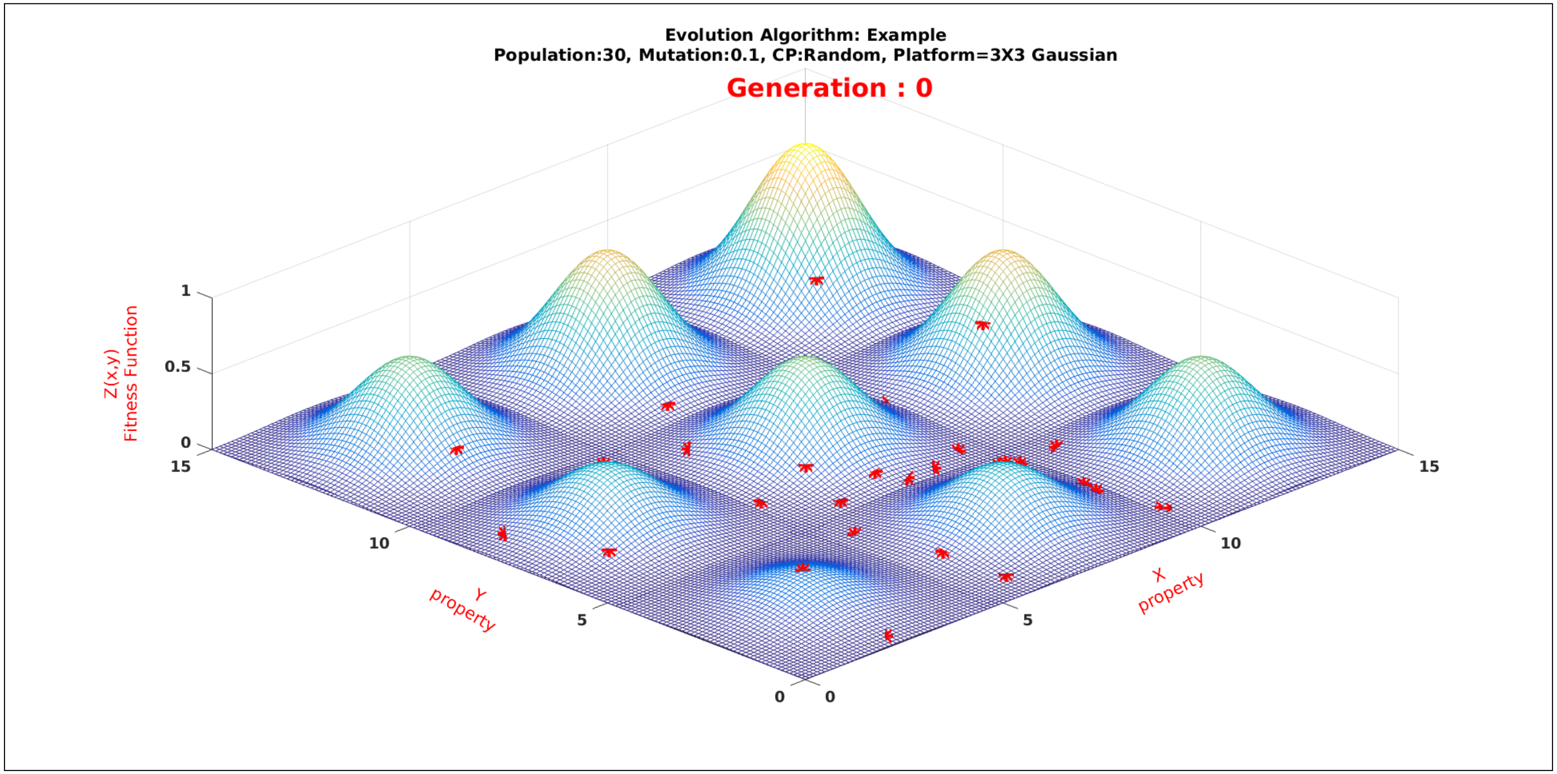
Figure: Cooperation schemes [10]



(a) without co-op. (b) with co-op.

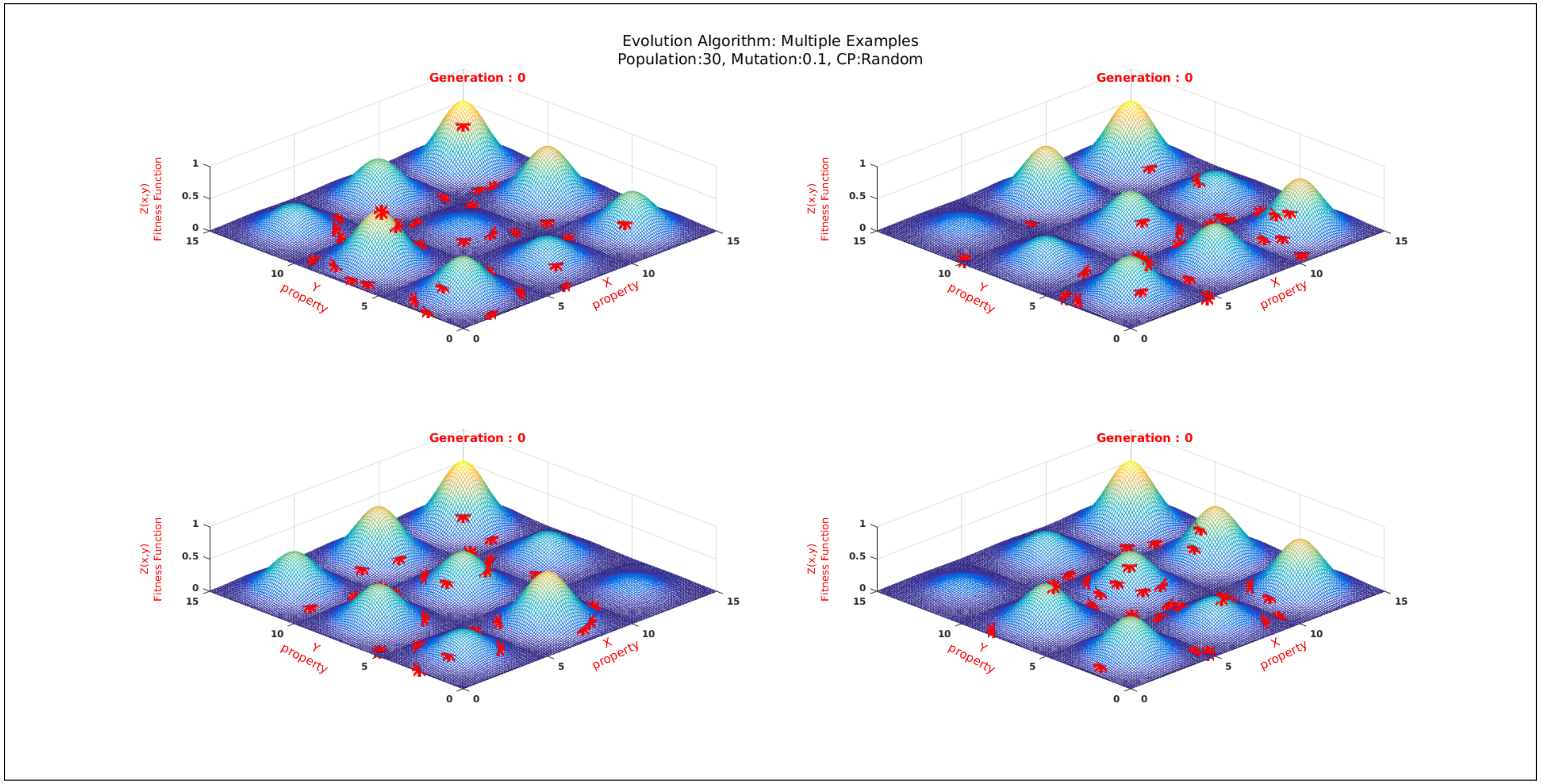
Figure: Cooperative co-evolution: an example

EA Examples



Video: EA realization on a landscape of multiple Gaussian functions

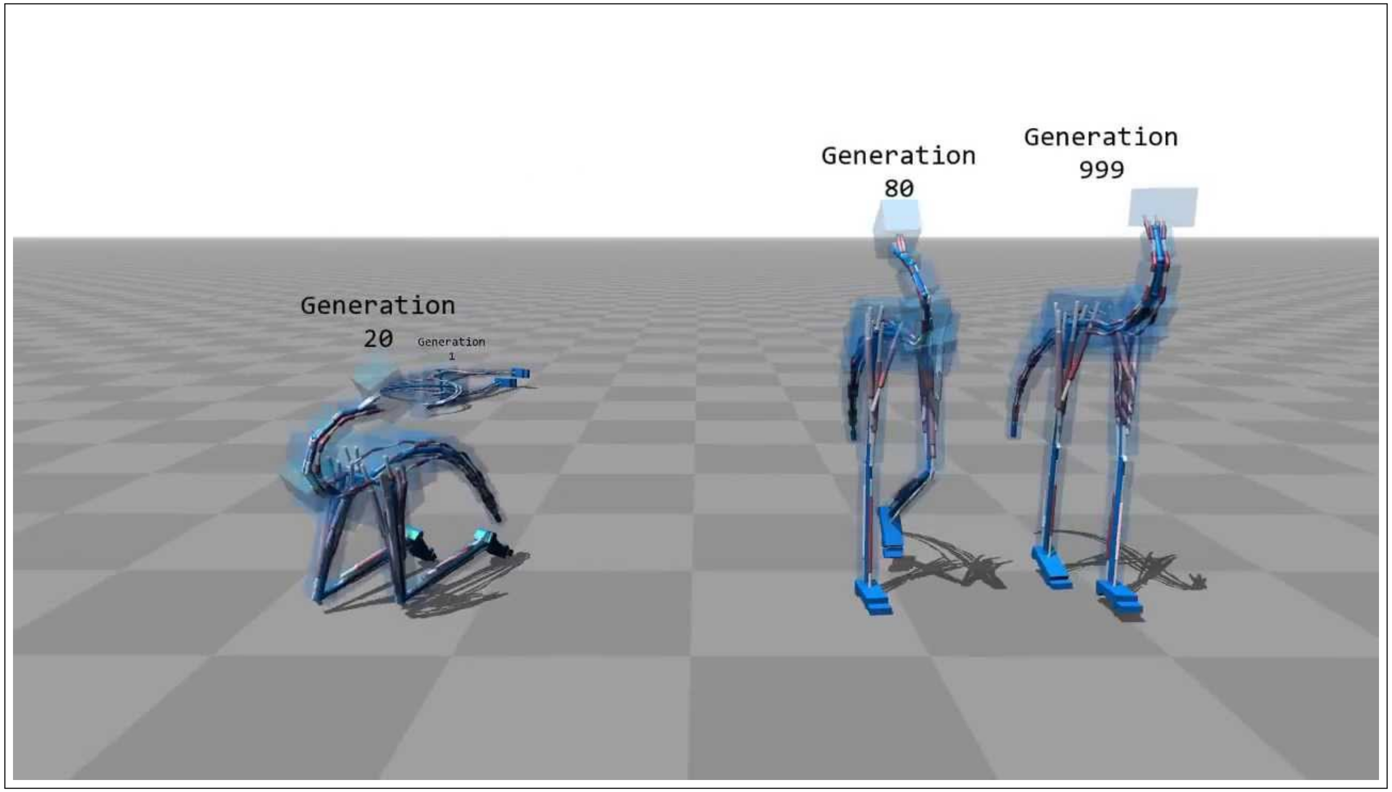
EA Examples



Video: Multiple runs of EA on different landscapes

Outline

Evolving Physical Design



Video: Evolving flexible locomotion for bipedal creatures [6]

Evolving Physical Design

- Space Technology 5 (ST5) is NASA's mission exploring earth's magnetic fields
- The planned nano-satellites orientation and altitude changed, thus requiring a new antenna design
- Instead of manually re-designing, NASA's Engineers developed an EA to design the first evolved antenna in outer-space

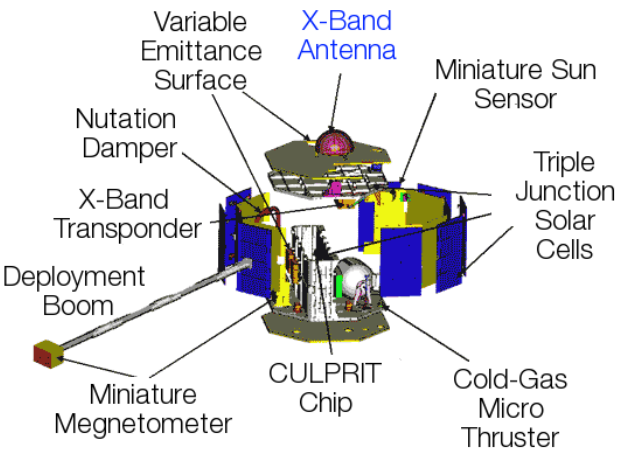


Figure: Spacecraft model

Figure: Best evolved antenna [7]

Evolving Physical Design

- RoboGen is an open source platform for evolving robots' physical designs

- It focuses on evolving easily manufactured robots with use of a small set of low-cost, off-the-shelf electronic components

Figure: RoboGen Framework [1]

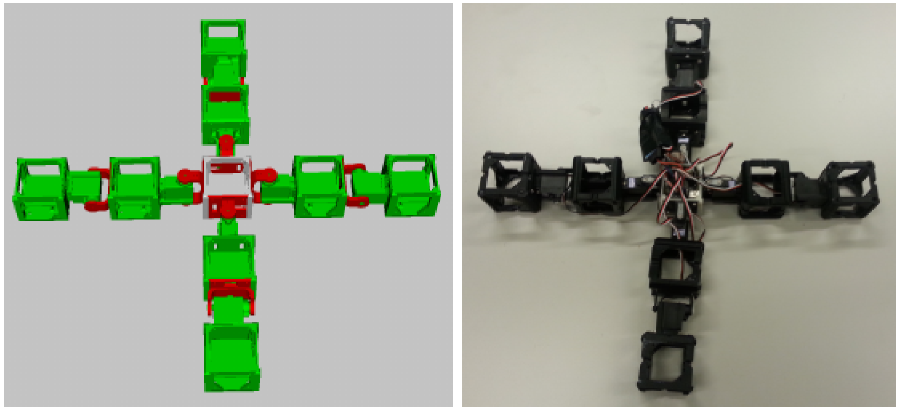
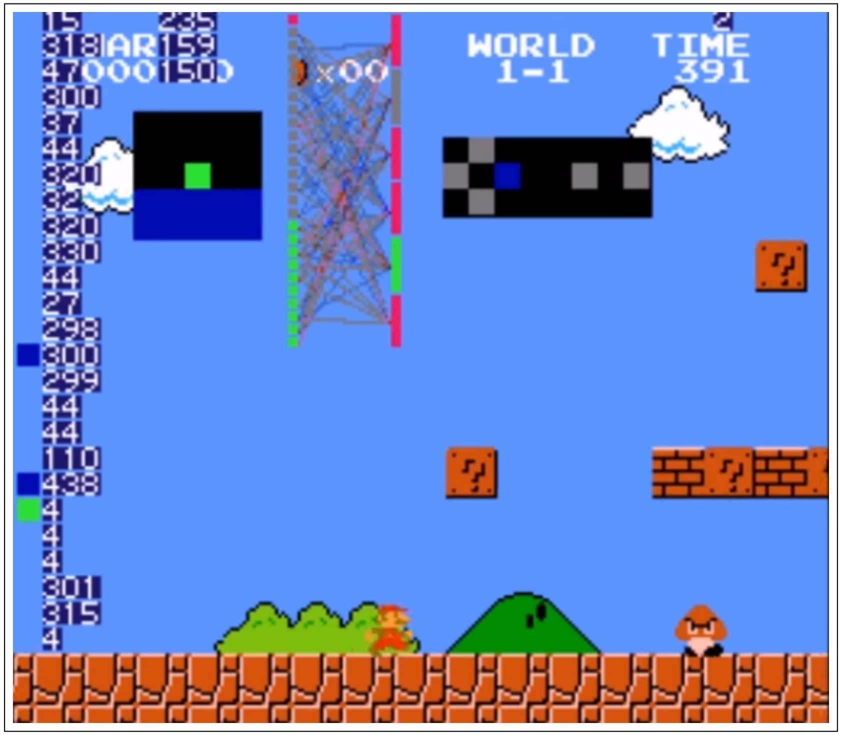


Figure: RoboGen: an example [2]

EA in Software



Video: Supermario using EA [14]

EA in Software

Algorithm	Citation
Evolutionary repair of faulty software	Arcuri, 2011
Co-evolutionary automatic programming for software development	Arcuri & Yao, 2010
MicroGP: An evolutionary assembly program generator	Squillero, 2005

Table: EA developed software examples

EA in Hardware

- Genotype-phenotype mapping is a crucial step in EA
- One idea is to map the genotype using lookup-table with a pre-assigned number to different circuit element types

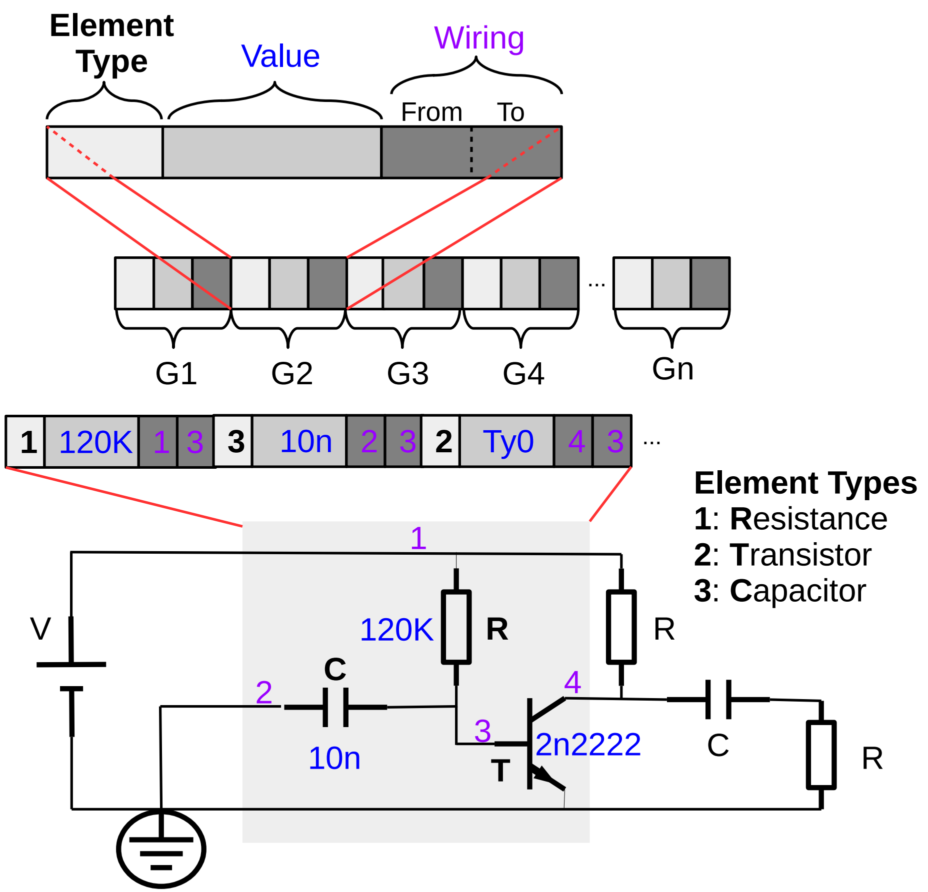


Figure: Analogue circuit genotype example [11]

EA in Hardware

- Like analogue circuits, digital circuits can be similarly mapped
- Encapsulation is possible, and different levels of granularity can be represented

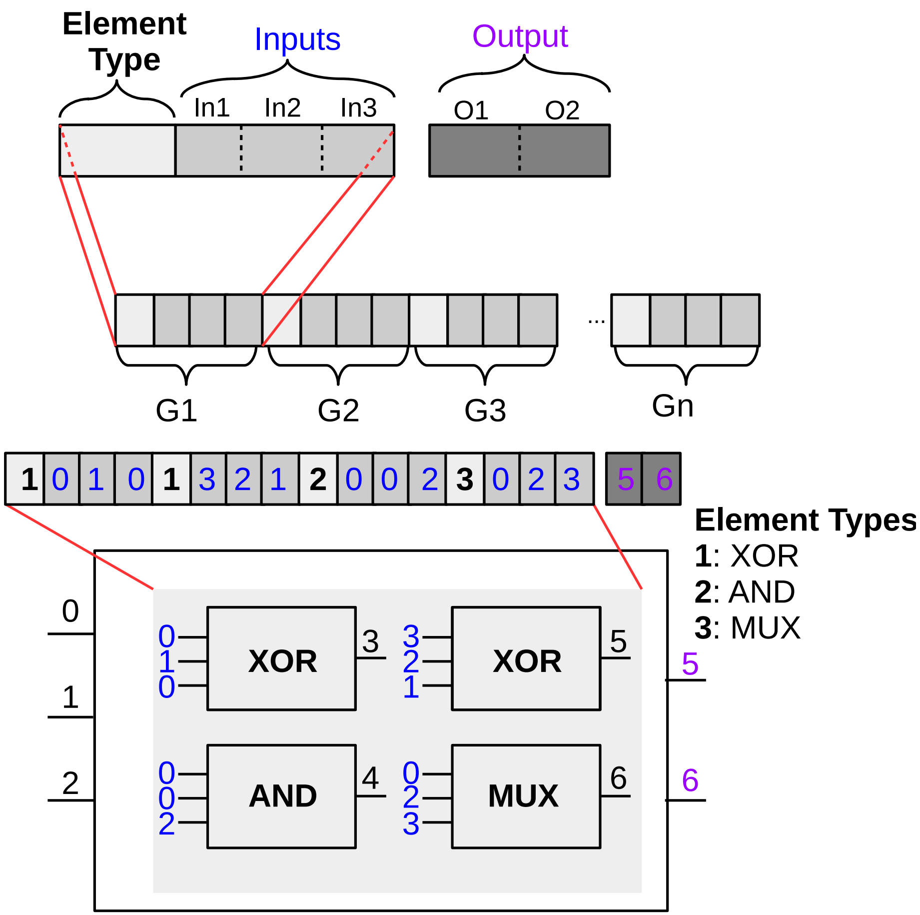


Figure: Digital circuit genotype example[11]

EA in Hardware

- Another way of genotype mapping is tree representation
- Different connection types can be mapped to alphabets

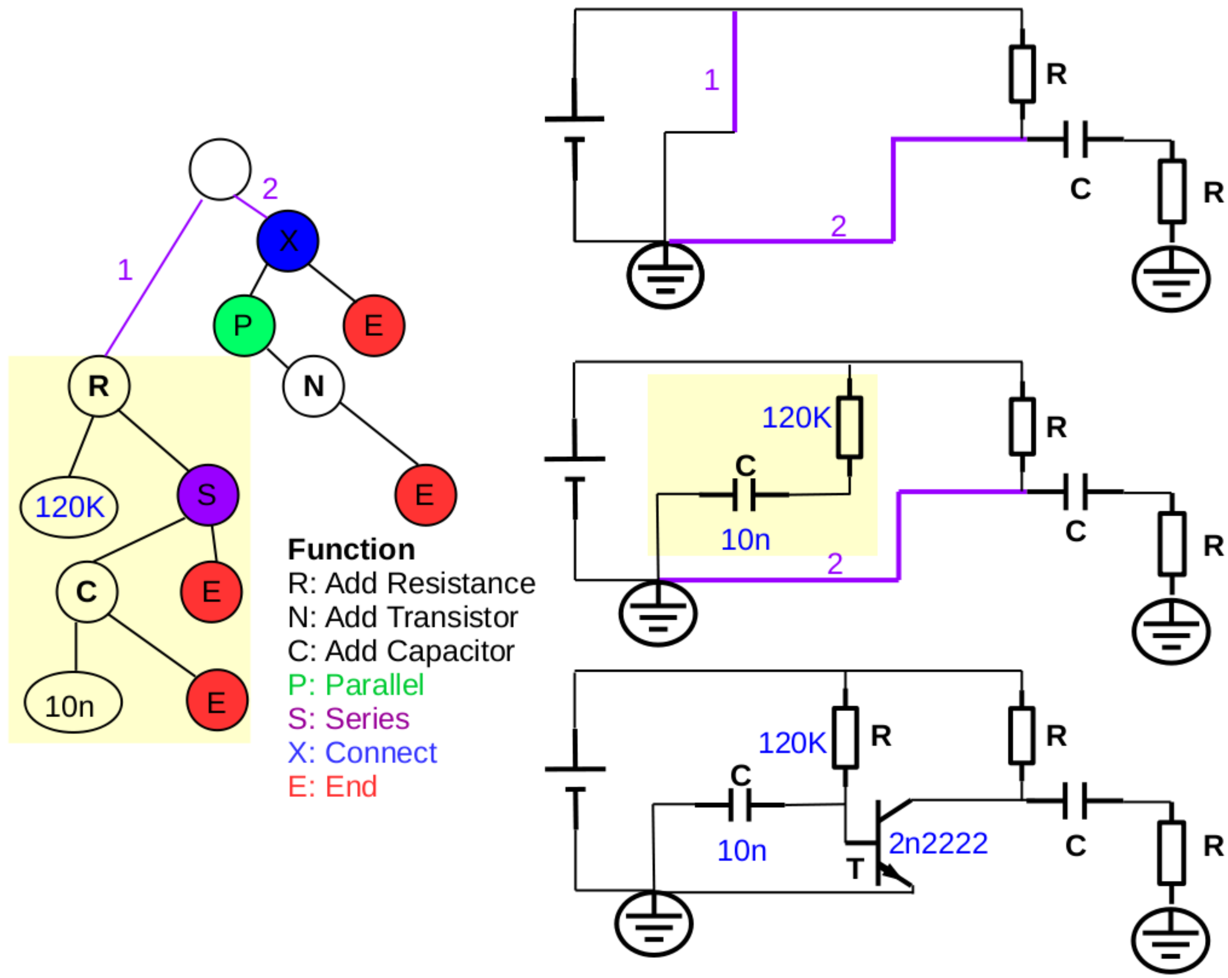


Figure: Tree gynotype representation

EA in Hardware

Invention	Date	Inventor	Patent
Low-voltage balun (balance/unbalance) circuit	2001	Sang Gug Lee	6,265,908
Mixed analog-digital circuit for variable capacitance	2000	Turget Sefket Aytur	6,013,958
Voltage-Current conversion Circuit	2000	Akira Ikeuchi Naoshi Tokuda	6,166,529
Low-voltage high current circuit for testing a voltage source	2001	Timothy Daun-Lindberg Micheal Miller	6,211,726
Low voltage cubic function generator	2001	Stefano Cipriani Anthony Takeshain	6,160,427
Tunable integrated active	2001	Robert Irvine Bernd Kolb	6,225,859

Table: List of patents re-invented using EA [8]

Outline

Conclusion

- Showed evolutionary algorithms' design axioms and demonstrated the algorithm using multiple examples
  - Perfect for problems that can't or are hard to be mathematically formalized
  - Single objective, multi-objective and objective free optimization
  - Trade-off between exploration and exploitation
  - Robust in dynamic environment
  - Cons: high computational cost as many evolutionary cycles are needed
- Highlighted use of Co-evolution
  - Co-evolution dynamics can be unpredictable (e.g. strategy recycling, dynamic fitness landscape, red queen effect)
- Demonstrated EA's uses in both physical design, software architecture and circuit design

References I

[1] Joshua E Auerbach and Dario Floreano.
Robogen - robot generation through artificial evolution.

[2] Joshua E Auerbach, Grégoire Heitz, Przemyslaw M Kornatowski, and Dario Floreano.
Rapid evolution of robot gaits.
In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*, pages 743–744. ACM, 2015.

[3] Leandro Nunes De Castro.
Fundamentals of natural computing: basic concepts, algorithms, and applications.
CRC Press, 2006.

[4] Kalyanmoy Deb.
Multi-objective optimization using evolutionary algorithms, volume 16.
John Wiley & Sons, 2001.

[5] Iztok Fister, Damjan Strnad, Xin-She Yang, and Iztok Fister Jr.
Adaptation and hybridization in nature-inspired algorithms.
In *Adaptation and Hybridization in Computational Intelligence*, pages 3–50. Springer, 2015.

References II

[6] Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen.
Flexible muscle-based locomotion for bipedal creatures.
ACM Transactions on Graphics, 32(6), 2013.

[7] Gregory S Hornby, Al Globus, Derek S Linden, and Jason D Lohn.
Automated antenna design with evolutionary algorithms.
In *AIAA Space*, pages 19–21, 2006.

[8] John R Koza, Martin A Keane, and Matthew J Streeter.
What's ai done for me lately? genetic programming's human-competitive results.
IEEE Intelligent Systems, (3):25–31, 2003.

[9] Joel Lehman and Kenneth O Stanley.
Abandoning objectives: Evolution through the search for novelty alone.
Evolutionary computation, 19(2):189–223, 2011.

[10] Laurent Lehmann and Laurent Keller.
The evolution of cooperation and altruism–a general framework and a classification of models.
Journal of evolutionary biology, 19(5):1365–1376, 2006.

References III

[11] Julian F Miller, Dominic Job, and Vesselin K Vassilev.
Principles in the evolutionary design of digital circuits—part i.
Genetic programming and evolvable machines, 1(1-2):7–35, 2000.

[12] Ian Millington and John Funge.
Artificial intelligence for games.
CRC Press, 2009.

[13] Jean-Baptiste Mouret and Jeff Clune.
Illuminating search spaces by mapping elites.
arXiv preprint arXiv:1504.04909, 2015.

[14] Michael Roberts.
Super mario bros. - neural network with genetic algorithm, July 2015.

[15] Moshe Sipper, Eduardo Sanchez, Daniel Mange, Marco Tomassini, Andrés Pérez-Uribe, and André Stauffer.
A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems.
Evolutionary Computation, IEEE Transactions on, 1(1):83–97, 1997.