ICCCI conference 2023

Comparing Lamarckian and Baldwinian Approaches in Memetic Optimization

Presenter: Mei Jiaojiao

Authors: Mei Jiaojiao, László Gulyás, János Botzheim

September 27, 2023

Eötvös Loránd University
Faculty of Informatics
Dept. of Artificial Intelligence

Table of contents

Introduction

Optimization

People optimize:

- Finding the best outcome with limited resources
- Obtaining the best education
- Securing the most decent job
- Finding the most desirable partner, etc.

Real-life problems:

- Optimization problems
- Complexity
- Often with only approximate solutions

Common optimization algorithms:

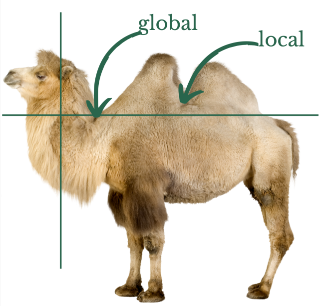
- Artificial Neural Networks(ANNs)
- Swarm Intelligence
- Numerical Optimization
- Evolutionary Algorithms, etc.

Continuous Optimization Problem

General Continuous Optimization Problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

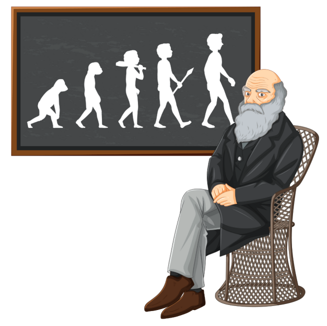
- **Local Minimum**: x^* in D where $f(x^*) \leq f(x)$ in a small neighborhood.
- **Global Minimum**: x^* in D where $f(x^*) \leq f(x)$ for all x in the domain D.

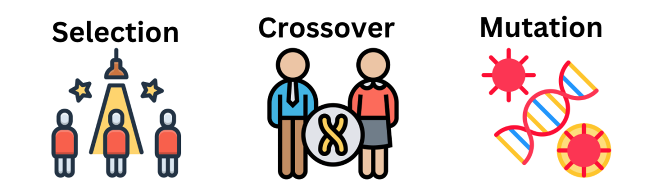


Genetic algorithm

Genetic Algorithm:

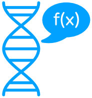
- Population-based optimization method
- Inspired by Darwinian theory of evolution
- Stochastic in nature
- Utilizes 3 principal bio-inspired operators

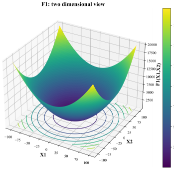


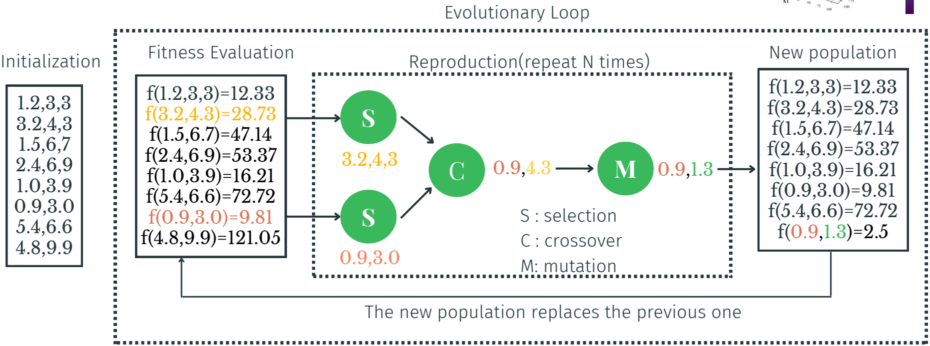


Genetic Algorithm

Each individual has DNA. DNA(genotype) is a vector.
Each vector x is a candidate solution to a function.







Memetic Algorithm

Memetic algorithm = Genetic algorithm + Local search procedure

- Individuals are capable of learning
- Learning from the local environment
- Learning leads to self-improvement
- Phenotype = Genotype + Local search procedure(learning)

Baldwinian Evolution

- "What is learned" passed on through genotypes to offspring
- Biological plausibility is questionable

Lamarckian Evolution

- "What is learned" passed on through phenotypes to offspring
- Biological plausibility is questionable

Why hybridize?

Genetic Algorithms

- Explore large, rough search spaces
- Difficulties with fine-tuning

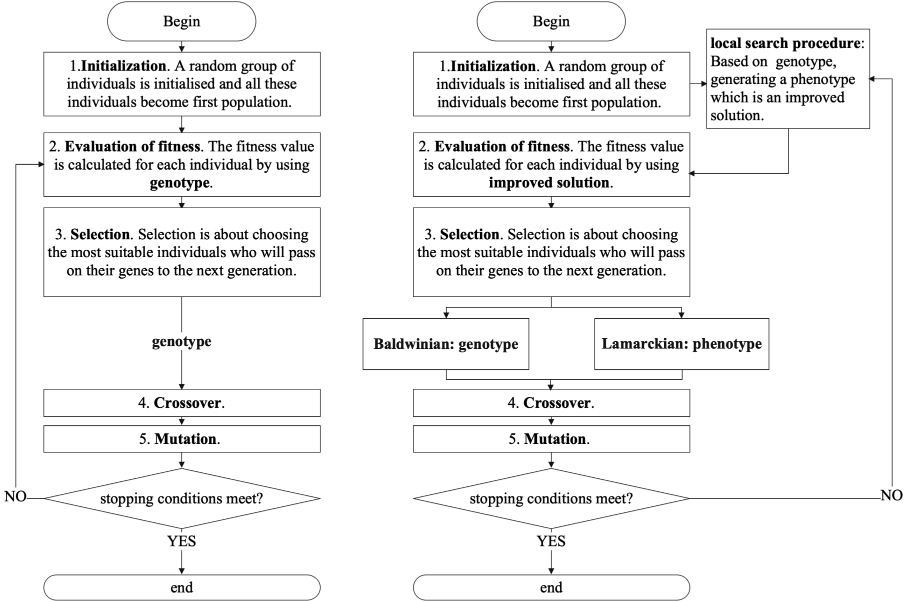
Local search techniques

- Optimize/converge fast
- Get stuck in local optima

Disadvantages

- There are costs of the learning
- The learning is not always good

Flowchart


Begin
1.Initialization. A random group of individuals is initialised and all these individuals become first population.
2. Evaluation of fitness. The fitness value is calculated for each individual by using genotype.
3. Selection. Selection is about choosing the most suitable individuals who will pass on their genes to the next generation.
genotype
4. Crossover.
5. Mutation.
stopping conditions meet?
NO
YES
end
Begin
1.Initialization. A random group of individuals is initialised and all these individuals become first population.
local search procedure: Based on genotype, generating a phenotype which is an improved solution.
2. Evaluation of fitness. The fitness value is calculated for each individual by using improved solution.
3. Selection. Selection is about choosing the most suitable individuals who will pass on their genes to the next generation.
Baldwinian: genotype
Lamarckian: phenotype
4. Crossover.
5. Mutation.
stopping conditions meet?
NO
YES
end

Contribution of this Work

Contribution 1

Evaluate the effectiveness of local search procedures

Contribution 2

Provide a comprehensive analysis of Baldwinian and Lamarckian algorithms using benchmark functions CEC-BC-2017

Methods

Selection Operators

Selection operators perform the following operations:

1. Select the best one or a fixed-percentage of individuals and classify them as **eligible**.
2. Remove an equal number of worst individuals and classify them as **non-eligible**.
3. The remaining individuals are considered **partners**.
4. Use the **eligible** individuals and **partners** to generate an equal number of new individuals.

Four selection operators are implemented:

- Steady State Genetic Algorithm
- Sorted Selection Part
- Sorted Selection All
- Roulette Wheel Selection

Crossover Operators

Single Point Crossover

$$\begin{aligned} Child_1 &= [P_1[1:c], P_2[c+1:D]] \\ Child_2 &= [P_2[1:c], P_1[c+1:D]] \end{aligned} \tag{1}$$

Two Point Crossover

$$\begin{aligned} Child_1 &= [P_1[1:c_1], P_2[c_1+1:c_2], P_1[c_2+1:D]] \\ Child_2 &= [P_2[1:c_1], P_1[c_1+1:c_2], P_2[c_2+1:D]] \end{aligned} \tag{2}$$

Probabilistic Crossover

$$Child_i = \begin{cases} P_1[i], & \text{if } r[i] \le C_r \\ P_2[i], & \text{otherwise} \end{cases} \tag{3}$$

Linear Combination Crossover

$$Child_i = C_r \cdot P_1[i] + (1 - C_r) \cdot P_2[i] \tag{4}$$

Mutation Operators and Local Search Operators

Mutation Operators:

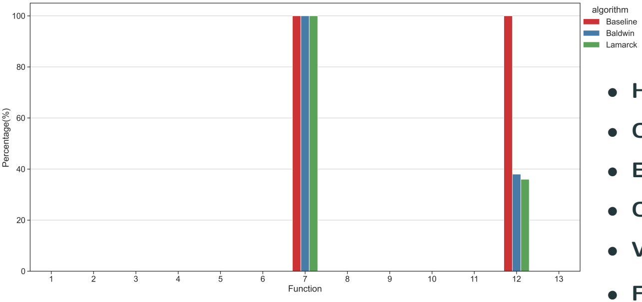
- Uniform Mutation:
  - **Iterate** over each gene in a solution
  - NewGene = OldGene + $U(-3 \cdot STD, 3 \cdot STD)$(**Controlled by** M_r)
  - $STD = R \cdot Domain$, where $Domain$ is the domain of a test function
  - R is a parameter controls the range of STD
  - A new solution is generated after each **complete iteration**
- Normal Mutation:
  - NewGene = OldGene + $N(0, STD^2)$

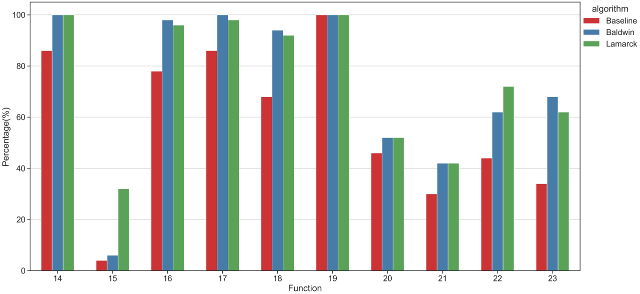
Local Search Operators:

- Generate a new solution using the mutation operator.
- If the new solution is better than the current solution, replace the current solution.
- Repeat this process multiple times.
- Return the best solution as the phenotype.

Results

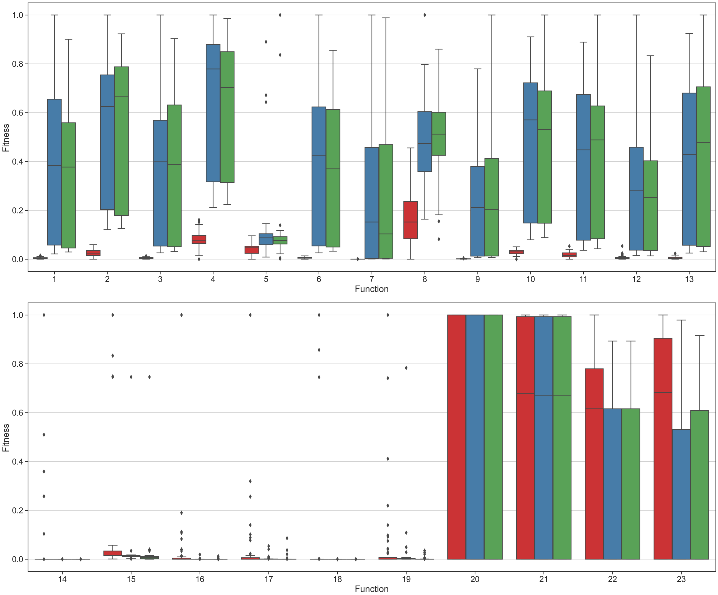
Success Rates in Finding the Global Optimum





- **H-axis**: different functions
- **Colors**: different algorithms
- **Each bar**: 50 solutions
- **Criterion**: $|\mathrm{s} - F(x^*)| \leq 0.0001$
- **V-axis**: percentage of success
- **Function Group**:
  - F1-F7: unimodal functions
  - F8-F13: multimodal functions
  - F14-F23: multimodal functions with fixed dimensions

Distribution of Best Solutions Found



- **H-axis**: different functions
- **Colors**: different algorithms
- **Each box**: 50 solutions
- **V-axis**: distribution
- **We expect**:
  - lower mean
  - less outliers

Conclusions

Conclusion 1

Local search procedures

- The baseline algorithm outperformed the Baldwinian and Lamarckian algorithms on function F12.
- All three algorithms achieved a success rate of 100 percent on function F7.
- None of the three algorithms were able to find the global minimum on functions F1-F6 and F8-F13 (success rate: 0 percent).
- The Baldwinian and Lamarckian algorithms outperformed the baseline algorithm on functions F14-F23.

Conclusion 1

Local search procedures were found to be effective.

Conclusion 2

Baldwinian vs. Lamarckian

- From the perspective of success rates:
  - The Lamarckian algorithm outperformed the Baldwinian algorithm on functions F15 and F22.
  - The Baldwinian algorithm outperformed the Lamarckian algorithm on functions F12, F16, F17, F18, and F23.
- From the distribution of results:
  - The Baldwinian algorithm outperformed the Lamarckian algorithm on functions F2, F3, F7, F9, F13, and F23.
  - The Lamarckian algorithm outperformed the Baldwinian algorithm on functions F1, F4, F5, F6, and F10-F12.

Conclusion 2

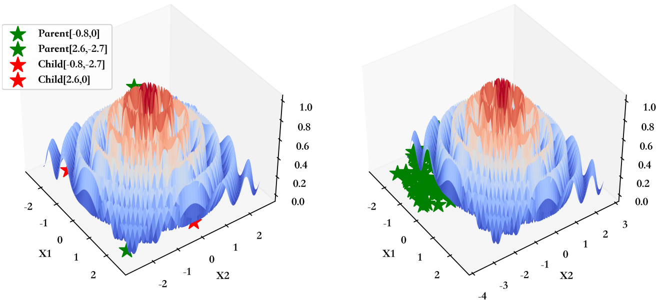
There is no clear winner between the Baldwinian and Lamarckian algorithms.

Future Direction

- Explore novel local search procedures
- Investigate algorithm parameter tuning, reduce number of evaluations
- Conduct comparative studies with other metaheuristic algorithms
- Validate the practical use through real-world case studies.

Future Direction: Make a jump

- At first, crossover allows you to jump
- As the evolution goes, everyone becomes more and more similar
- Then the population will stuck in a local minimum
- In this case, crossover won't help you, local search won't help you....
- Mutation can help you but with a very low probability



Maybe try to make a jump...... when stuck in local minimum?

Questions?

- **Mei Jiaojiao** : v6fisc@inf.elte.hu
- **László Gulyás** : lgulyas@inf.elte.hu
- **János Botzheim** : botzheim@inf.elte.hu

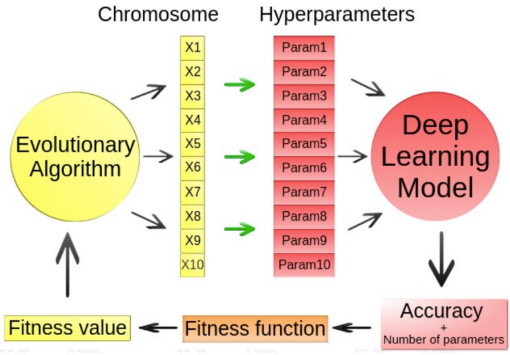
Backup slides: Network Routing Optimization

- **Problem**: Optimize data routing in communication networks.
- **Chromosome Representation**: Genes represent route paths (nodes and links).
- **Fitness Function**: Evaluate routes based on latency, bandwidth utilization, and packet loss.
- **Genetic Operations**: Apply crossover and mutation to discover efficient routing solutions.

Backup slides: Evolutionary NNs

Hyperparameter optimization of neural networks

- Structure parameters: Number of neurons, Number of layers, Types of layers, Number of convolution kernels
- Backpropagation parameters: Learning rate, Momentum
- Regularization parameters: Dropout probability, L1/L2 regularization



Backup slides: Basic concepts

- **Gene**: functional entity that encodes a specific feature/property of the individual (e.g. hair color)
- **Allele**: value of gene (e.g. blonde)
- **Genotype**: the specific combination of alleles carried by an individual(DNA)
- **Phenotype**: the physical makeup of an organism(Body)
- **Locus**: position of the gene within the chromosome
- **Individual**: chromosome, represents a candidate solution for the problem
- **Population**: collection of individuals currently alive
- **Fitness value**: the individuals are evaluated according to some criterion how good solution they can provide to the given problem
- **Meme**: meme = unit of cultural transmission (the "genes" of cultural evolution) (Dawkins: The selfish gene, 1976)
- **mimema**: imitation
- **Local search**: lifelong learning

Backup slides: Local search

Local search

- **global workspace**: current node with its small environment
- **searching rule**: in each step the current node is exchanged for better child
- **control strategy**: use an evalution(objective, fitness, heuristic) function to select a better child

Example of local search: Hill climbing method(implementation is easy), some improvements:

- several current nodes(local beam search)
- several attempts(random-restart search)
- give up the greedy strategy(simulated annealing)
- recognize smaller cycles(tabu search)

When is local search worth using? There is no chance to find a solution without strong heuristics.