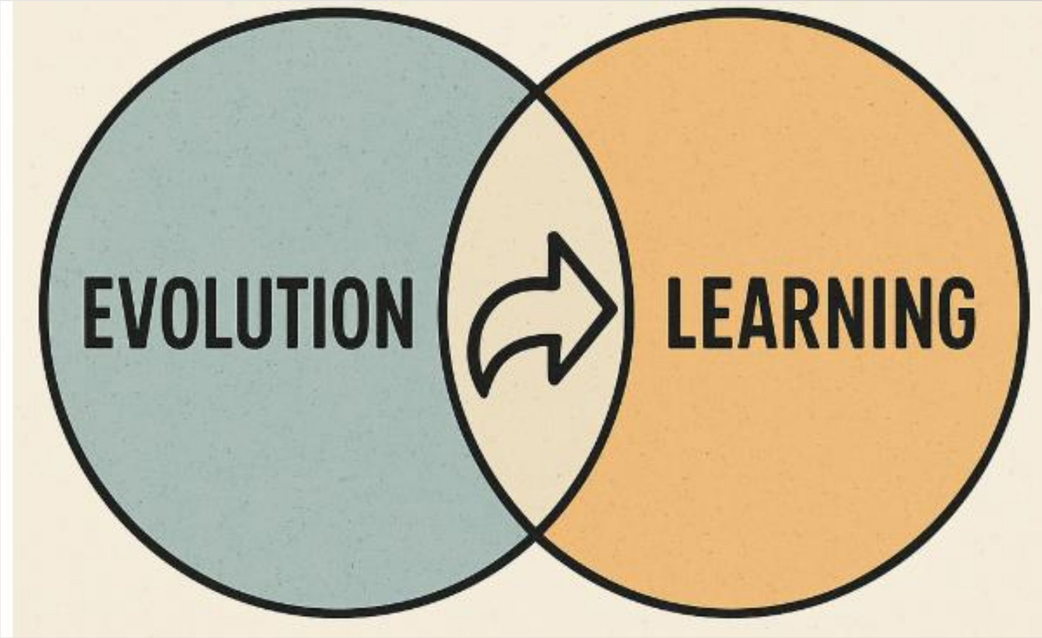
AI LAB 2025

AN INTRODUCTION TO ARTIFICIAL INTELLIGENCE

SHAY BUSHINSKY, SPRING 2025

LAB2: GENETIC PROCESSES

EVOLUTION AND LEARNING



IN THIS LECTURE

- Baldwinism vs. Lamarckism
- Experiments in the Baldwin Effect
- Memetic Algorithms (MA)
- Culture Algorithms (CA)
- Cultural Transmission

GENETIC LEARNING CAPABILITY

- James Baldwin (19th-century psychologist)
- Studied learning as an evolutionary advantage
- Key question: How does learning enhance survival?
- Idea: Learning can shape evolution (Baldwin Effect)

GENETICS VS. LEARNING

- Example: Birds avoiding toxic insects
- Genetic: Avoidance is innate
- Learned: Avoidance is from painful experience
- Learning is flexible but risky and costly
- Genetics offers safer, faster response

TRADE-OFFS & BENEFITS

- Learning aids adaptation to changing environments
- Genetics suits stable, repeatable threats
- Learning requires energy and carries risk
- Evolution may favor learning when survival depends on it

LAMARCKIAN EVOLUTION

- Proposed in 1809 by Jean Baptiste Lamarck
- Traits acquired during life are inherited
- Focused on phenotypic changes
- Example: long legs & webbed feet in birds

Jean Baptiste Lamarck
1744-1829

LAMARCK'S THEORY OF EVOLUTION

- Use & Disuse: Traits not used are lost
- Useful traits are strengthened
- Inheritance of acquired traits across generations

LAMARCKISM EXAMPLES

- Giraffes stretch necks → longer-necked offspring
- Blacksmith builds muscle → muscular sons
- Known as 'soft inheritance'

LAMARCK'S CONJECTURE

- Traits from experience are passed on
- Offspring gain benefits of learned traits
- Darwin considered Lamarck's views, but preferred natural selection

DARWINIAN EVOLUTION

- Published 'The Origin of Species', 1859
- Acquired traits are not inherited
- Evolution via natural selection:
- – Mutation in genotype
- – Selection on phenotype

Darwin
1809-82

NATURAL SELECTION

- Survival depends on phenotype differences
- Contrasts with artificial (human-driven) selection
- Better adapted individuals reproduce more
- Traits linked to survival are selected over time

NATURAL SELECTION (EXPANDED)

- Individuals vary due to genetic differences
- Better-adapted traits improve survival & reproduction
- Advantageous genes passed to next generation
- Over time: beneficial traits accumulate
- May result in adaptation or new species

ARTIFICIAL SELECTION

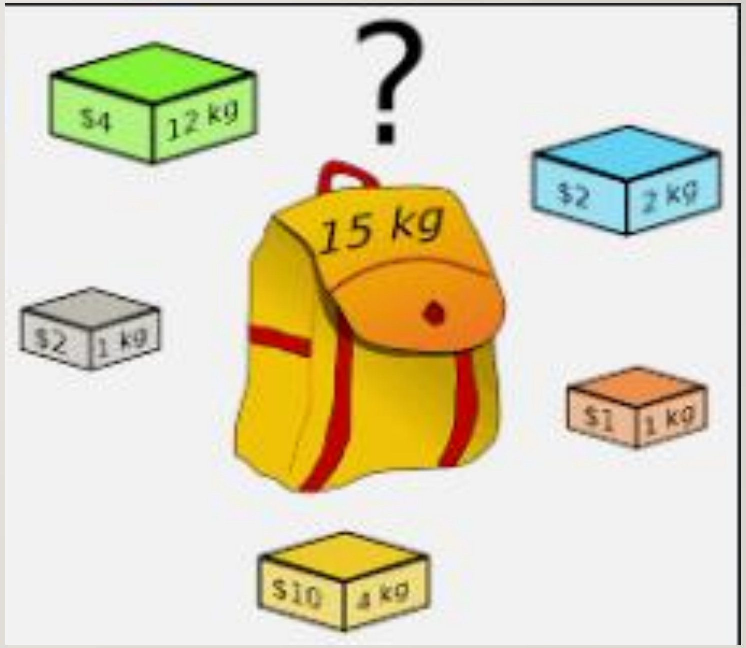
- Humans select traits for breeding
- Directed goal (e.g., yield, looks)
- Examples: crops, pets, livestock
- Unlike natural selection: not environment-driven
- Driven by human intervention

PHENOTYPE VS. GENOTYPE

- Genotype: An organism's gene set
- Phenotype: Observable traits
- Phenotype = Genotype + Environment
- Genotype differences → different phenotypes

AI EXAMPLE: THE KNAPSACK PROBLEM

- Genotype: Binary string (1 = item included, 0 = not)
- Phenotype: Actual set of selected items
- Interpreted from genotype under capacity limits



THE KNAPSACK PROBLEM

- **Genotype**: A binary string, where each bit represents whether an item is included in the knapsack (1) or not (0)

- **Phenotype**:
  - **The actual set of items** selected for the knapsack, decoded by interpreting the binary string and selecting items corresponding to the bits set to 1, while considering the knapsack's capacity constraint

THE ENVIRONMENT IN THE KNAPSACK PROBLEM

In evolutionary algorithms applied to the knapsack problem, the **environment** includes:

1. **Knapsack capacity**
   → Determines whether a solution (phenotype) is **valid or penalized**.
2. **Item values and weights**
   → Define the **fitness landscape**: how good each genotype is.
3. **Penalty strategies / repair mechanisms**
   → Influence whether and how an overweight solution is corrected or penalized.
4. **Selection pressure / competition**
   → Only the best phenotypes (based on fitness) are likely to reproduce.

THE BALDWIN EFFECT

- Proposed by James Mark Baldwin (1896)
- Learning behavior can guide evolution
- Learned traits increase fitness → genetic support
- Shifted aside after Mendelian genetics rediscovery (1900)

James Mark Baldwin
1861-1934

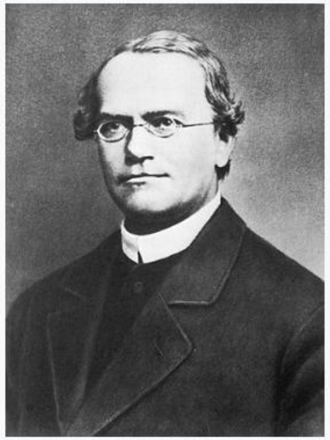
MECHANISM OF THE BALDWIN EFFECT

- Learned behavior boosts survival
- Selection favors learners
- Genes that support learning become common
- Eventually, behavior becomes instinctive

BALDWIN VS. MENDELIAN GENETICS

- Mendel: traits passed via genes
- 1900: focus shifted to genetic determinism
  - How traits are passed from one generation to the next
- Baldwin's theory seen as complex and indirect
- Overshadowed by simpler gene-focused models

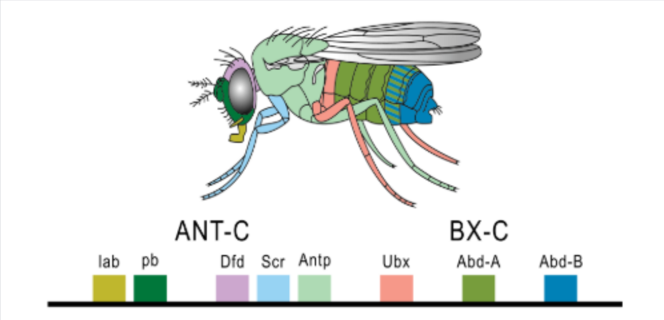
Gregor Mendel
OSA



Born	Johann Mendel 20 July 1822

MODERN RELEVANCE OF THE BALDWIN EFFECT

- Resurgence in evo-devo, epigenetics
- Recognizes learning's evolutionary role
- Combines genetics, behavior, and environment
- Now part of integrative evolutionary theory



Expression of all 8 Hox genes in the fruit fly *Drosophila melanogaster*

BIRDS & THE BALDWIN EFFECT

- Birds learn to avoid toxic insects
- Learning allows some to survive initially
- Genes that aid learning selected for
- Over time: instinctual avoidance emerges

THE KEY ASPECTS OF BALDWINISM

1. Learned behaviors in response to environment
2. Learning affects survival and reproduction
3. Selection favors learning predisposition
4. Genetic evolution interacts with plasticity

CONFIRMATION OF THE BALDWIN EFFECT

THE HINTON NOWLAN EXPERIMENT (1987)

Steven Nowlan and Geoffrey Hinton, renowned as the "Godfather of AI"

CONFIRMING THE BALDWIN EFFECT

- Demonstrated that:

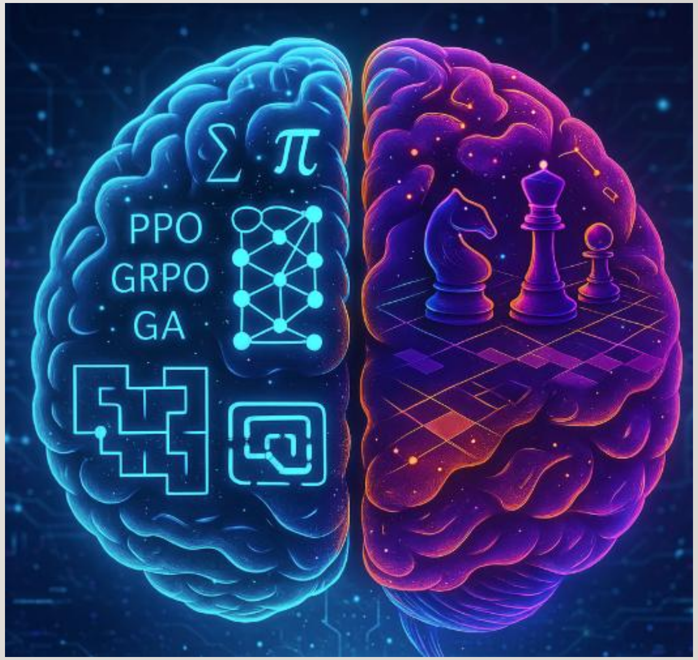
- "Learning alters the shape of the search space in which evolution operates and thereby provides good evolutionary paths"

EXPERIMENT PRINCIPLE

- Change fitness without changing genetic structure
- Achieved via local search

THE EXPERIMENT SETUP

- Simulated 'neural network' with 20 connections
- Each individual = 010010...
- Goal: evolve into target network
- The target: an arbitrary 20-length bit string
- Each had 1000 attempts



POPULATION OF CREATURES

- 20 alleles per organism (0, 1, or ?)
- Net only works in one exact form
- Without learning: GA ≈ random search

EXPERIMENT PARAMETERS

- 1000 population, replaced each generation
- Alleles initialized: 50% ?, 25% incorrect, 25% correct
- Each performs 1000 learning trials

TRIAL-AND-ERROR LEARNING

- Each iteration: 1000 trials
- 1. Copy fixed bits
- 2. Fill '?' with random bits
- 3. Compare with target
- 4. Stop if match found

COMPUTING FITNESS

- Count unused trials n
- If match in 300 trials, $n = 700$
- Fitness: $f = 1 + (19n / 1000)$

LEARNING EFFECTS ON SELECTION

- Parents selected based on fitness
- Probability: $1 + (19n / 1000)$
- $n=1000 \rightarrow$ 20x more likely to reproduce
- RWS favors faster learners

FITNESS COMPUTATION EXAMPLES

- Let n = trials left unused from 1000 attempts
- Fitness = 1 + (19 * n / 1000)
- – n = 1000 ⇒ f = 20 (fast learner)
- – n = 600 ⇒ f = 11.4 (moderate learner)

CORRECT CONNECTION ANALYSIS

- Target = '000111', String = '0??011'
- Correct fixed = 3/6 = 50%
- Incorrect fixed = 1/6 = 16.7%

OBSERVED BALDWIN EFFECT

- After 50 generations:
- – Correct connections rose to 60%
- – Incorrect dropped to 0%
- – Survival increased to 90%

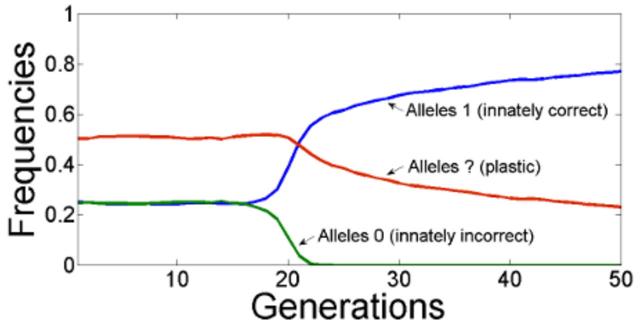


Fig. 1. Replication of Hinton and Nowlan's original simulation. The population is able to quickly learn the solution as innately incorrect alleles 0 are eliminated from the population, although the frequency of plastic alleles ? remains relatively high.

VIRTUAL BALDWIN EFFECT

- '?' (learnable connections) decreased
- Indicates evolution toward hardwired behavior
- => Learning facilitated genetic evolution

WHY ARE RANDOM TRIALS CONSIDERED LEARNING?

- Wildcard genes (e.g., '?') simulate flexibility
- Individuals test different values during lifetime
- Successful configs boost fitness
- Evolution favors genotypes that learn better

EXPERIMENT SUMMARY

- Hinton & Nowlan confirmed Baldwin Effect
- Simulated trial-and-error learning
- Showed learning can guide evolution
- Fast learners more likely to pass on genes

THE BALDWIN EFFECT: BUYING TIME

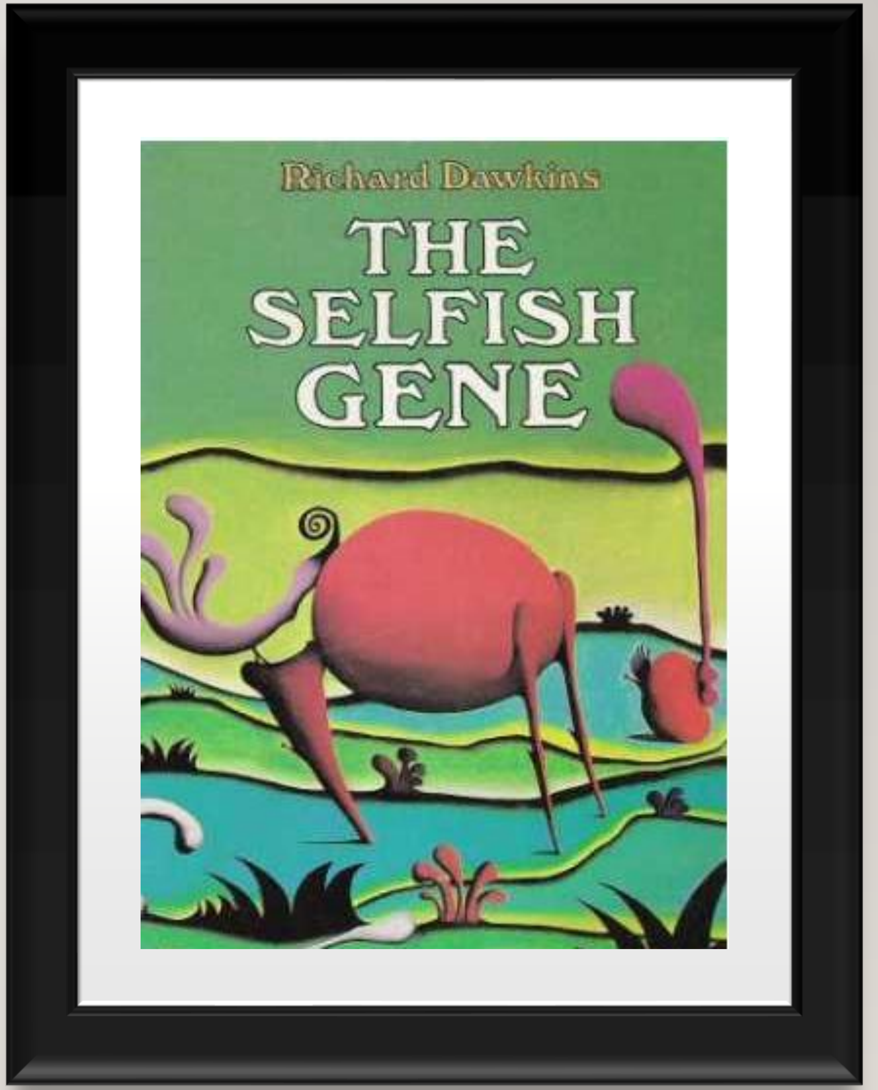
- Phenotypes may lead; genes follow.
- Learning/plasticity help survival in new environments.
- These traits maintain viability until genetic adaptation.
- Natural selection eventually encodes the traits.
- Still debated, but influential for explaining complex traits.

MEMETICS

The evolution of culture

CULTURAL INHERITANCE

- Richard Dawkins introduced **“meme”** in *The Selfish Gene* (1976)
- Defined as a unit of cultural transmission or imitation
- Memes spread by non-genetic means



WHAT IS A MEME?

- Cultural parallel to genes
- Passed via imitation
- Chosen to rhyme with “gene”
- Basis of cultural evolution

MEMETIC VS GENETIC EVOLUTION

- **Genes:** passive carriers in GAs
- **Memes:** active agents of change
- Used in memetic algorithms

MEMETIC ALGORITHMS

- Combine evolution with learning
- Balance generality and specificity
- Premise: evolution not limited to genes
- Leverages Baldwin Effect: learned behaviors aid adaptation

MA VS. GA

Mechanism	Memetic	Genetic
Evolution	Culture	Organic
Individuals	Active (Meme)	Passive (Gene)
Operators	Mutation/Selection/ Transmission	Mutation / Selection
Operator scope	Problem specific	Blind
Learning	Iterative Improvement	None
Selection	Fitness + Baldwinian	Fitness Based

COMMON ALGORITHM TYPES

1. Hybrid Evolutionary Algorithm
2. Baldwinian Evolutionary Algorithms
3. Lamarckian Evolutionary Algorithms
4. Genetic Local Search
5. Cultural Algorithms

LEARNING AND EVOLUTION

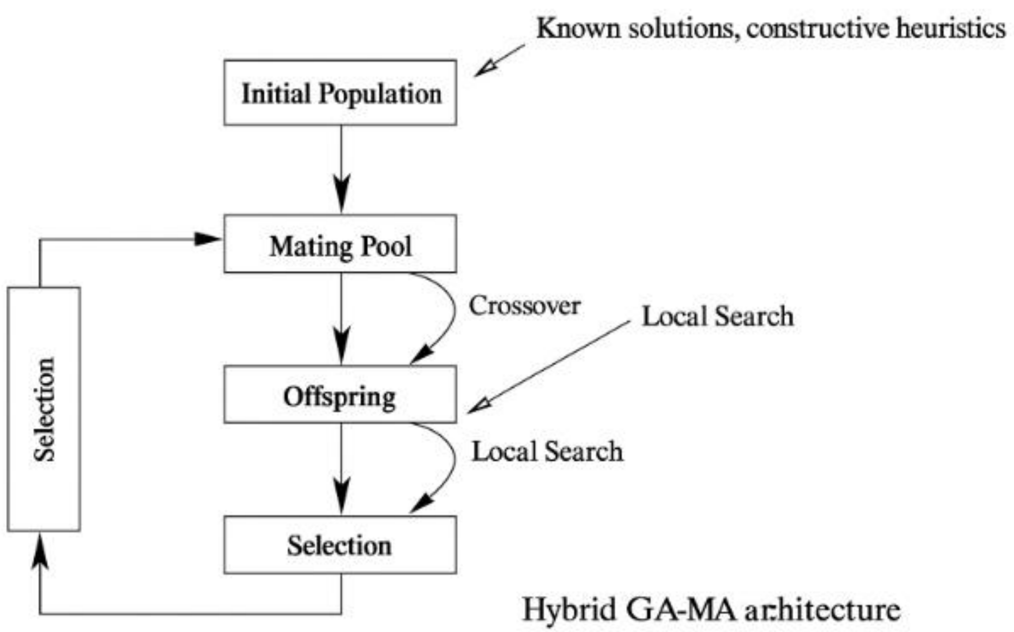
- Learning shapes fitness landscape
- **Baldwinian:** learning improves fitness only
- **Lamarckian:** learning changes both genotype and fitness

PURE MEMETIC ALGORITHM FLOW

- Initialize population
- Evaluate individuals
- Evolve new generation
- Select for local improvement
- **Apply:**
  - Meme-based learning
  - Baldwinian or Lamarckian refinement

HYBRID GA-MA ARCHITECTURE

- GA framework enhanced with local search
- Local improvements guide gene refinement
- **Structure:**
  - Population → Mating Pool → Crossover/Mutation → Local Search → Offspring



Hybrid GA-MA arhitecture

MA PARAMETERS

MEMETIC ALGORITHM CONTROL PARAMETERS

FREQUENCY AND INTENSITY

- Defines balance between evolution (exploration) and learning (exploitation)
- Frequency = how often learning occurs
- Intensity = how deep learning modifies the solution

POPULATION SUBSET

- Portion of population undergoing learning
- Subset choice affects MA efficiency
- Strategies:
- - Fitness-based
- - Distribution-based

MEME SELECTION

- Choose learning procedure suited to problem
- Each meme favors different neighborhoods
- Selection depends on problem characteristics

COMBINATORIAL OPTIMIZATION MEMES

- Memes are heuristics tailored to problem
- Examples:
- - k-gene exchange
- - edge exchange
- - first-/best-improvement

K-GENE EXCHANGE

- Swap k genes within or between individuals
- Explores local solution space
- Useful for problems like TSP

EDGE EXCHANGE

- Swap edges to enhance solution
- Example: 2-opt heuristic in TSP
- Refines genetic outcomes

FIRST VS. BEST IMPROVEMENT

- First-improvement: pick first better neighbor
- Best-improvement: search all, pick best
- Tradeoff: speed vs. quality

MA FUNCTIONAL DRILLDOWN

The Memetic Algorithm Blocks

GENERATE INITIAL POPULATION

- Loop from 1 to population size
- Generate new solution
- Apply memetic improver
- Return population

RESTART POPULATION

- Select top individuals to preserve
- Fill rest with new/improved solutions
- Return new population

DO GENERATION

- • Select individuals from population
- • Apply memetic operators
- • Update population with improved solutions

MEMETIC IMPROVER

- Repeat until termination:
- - Apply memetic operator
- - Replace if improved
- Return best individual

WHICH MEMETIC OPERATOR?

- Operator = learning method
- Chosen based on problem domain
- Example: Tabu Search is better than Simulated Annealing for Graph Coloring
- May vary across evolution

MA OPERATORS

- Operate on many genes or full population
- Types:
- 1. Recombination – combine k agents into one
- 2. Local search – refine individual

PROBLEM WITH TRADITIONAL CROSSOVER

- Classical crossover is blind
- Mixes traits without compatibility
- Can disrupt good traits

Avoid Spoiling Good DNA

- Solution encoding should support evaluation of sub-traits
- Enables preserving valuable genetic material

RECOMBINATION OPERATORS

- Designed to reduce disruption
- Use domain knowledge
- Transfer relevant traits to next generation

MEMETIC OPERATOR FOR TSP

- Avoid disrupting optimal subpaths
- Combine useful contributions from both parents

DOMAIN-SPECIFIC OPERATORS

- **TSP**:
- - Order Crossover (OX): preserves city order
- - PMX: avoids duplicates
- **Job Scheduling**:
- - JOX: preserves job constraints
- **VRP**:
- - Route-based crossover for feasible routes
- **Function Optimization**:
- - BLX-α: blends values within useful ranges

SUMMARY OF THE CLASSIC TSP CROSSOVER OPERATORS

Operator	Description
Order Crossover (OX)	Preserves relative city order from parents.
Partially Mapped Crossover (PMX)	Maintains city uniqueness by mapping sections.
Cycle Crossover (CX)	Preserves position by identifying cycles between parents.
Edge Recombination (ER)	Uses edge table to preserve adjacency information.

POSITION-BASED CROSSOVER (PBX)

- **How it works:**
  - Random positions are chosen and fixed from one parent.
  - Remaining cities are inserted in order from the second parent.
- **Preserves:** Absolute positions of selected cities.
- **Use case:** Useful when some city positions are highly beneficial.

EDGE ASSEMBLY CROSSOVER (EAX)

- **How it works:**
  - Builds edge sets from both parents.
  - Constructs intermediate graphs (AB-cycles) and extracts improved tours.
- **Preserves:** High-quality edges and tour segments.
- **Strength:** One of the most effective TSP crossovers; used in advanced heuristics.

GREEDY SUBTOUR CROSSOVER (GSX)

- **How it works:**
  - Begins with a city and greedily chooses the next city based on shortest unused edge from either parent.
- **Preserves:** Low-cost connections and greedy subtours.
- **Advantage:** Local optimization behavior within crossover.

DISTANCE PRESERVING CROSSOVER (DPX)

- **How it works:**
  - Constructs offspring using common edges from parents.
  - Fills gaps with nearest-neighbor heuristics.
- **Preserves:** Edge distance structure.
- **Goal:** Maintain structural similarity with parents.

HEURISTIC CROSSOVER (HX)

- **How it works:**
  - Uses heuristics like nearest neighbor or cheapest insertion to guide crossover.
- **Customizable:** Often problem-specific (e.g., integrating traffic patterns).
- **Strength:** Tailored to practical constraints in real-world TSP variant

JOB SCHEDULING: JOB ORDER CROSSOVER (JOX)

- **Purpose:** To preserve critical job execution sequences while combining solutions.
- **How it works:**
  - Selects a subsequence of job orders from one parent.
  - Fills in the rest from the second parent while maintaining the relative order.
- **Why it matters:**
  - Prevents violation of job precedence or resource constraints.
  - Enhances feasibility and quality in job-shop and flow-shop scheduling problems.

VEHICLE ROUTING PROBLEM (VRP): ROUTE-BASED CROSSOVER

- **Purpose:** To create feasible and efficient delivery routes.
- **How it works:**
  - Extracts entire routes (or route segments) from each parent.
  - Combines them while avoiding duplicate customers and ensuring capacity constraints.
- **Why it matters:**
  - Maintains route validity and avoids customer repetition.
  - Retains beneficial structure of existing high-quality routes.

FUNCTION OPTIMIZATION: BLEND CROSSOVER (BLX-Α)

- **Purpose:** To explore the solution space around parent values.
- **How it works:**
  - For each gene, generates offspring values from a range defined by the parents.
  - Range = $[\min - \alpha d, \max + \alpha d]$ where $d = |\text{parent1} - \text{parent2}|$
- **Why it matters:**
  - Encourages controlled exploration beyond parent values.
  - Ideal for real-valued parameter tuning and continuous optimization.

SUMMARY

- Domain-specific recombination avoids disruption
- - Ensures key traits are preserved
- - Boosts memetic algorithm performance

CULTURAL TRANSMISSION

A population based inheritence

CULTURAL ALGORITHM (CA) OVERVIEW

- Evolutionary algorithm with cultural knowledge.
- Dual inheritance: population and belief space.
- Enhances learning and diversity.

APPLICATIONS

- Engineering design
- Scheduling
- Resource allocation
- Machine learning
- Multi-objective optimization

BENEFITS OF CA

- Efficiency: Belief space accelerates search.
- Adaptability: Learns from history.
- Diversity: Balances exploration & exploitation.

KEY COMPONENTS

- Population Space: Evolves via selection, crossover, mutation.
- Belief Space: Stores shared knowledge (norms, values).
- Interacts to guide search process.

CA WORKFLOW

- 1. Initialize population and belief space.
- 2. Evaluate population using fitness.
- 3. Update belief space from best individuals.
- 4. Influence individuals using belief space.
- 5. Apply genetic operators.
- 6. Repeat until convergence.

INFLUENCE FUNCTION IN CA

- Guides evolution using cultural knowledge.
- Connects belief space to population.
- Steers search toward promising areas.

TYPES OF KNOWLEDGE IN BELIEF SPACE

- Normative: Acceptable ranges/behaviors.
- Situational: Examples of success/failure.
- Domain-Specific: Expert/practical insights.
- Temporal: Trends over generations.
- Spatial: Search space topography.

INTEGRATING BELIEF SPACE KNOWLEDGE IN CULTURAL ALGORITHMS

- Situational Knowledge:
- - Stores examples of top-performing solutions.
- - Injects successful patterns into future individuals (e.g., elite preservation).

INTEGRATING BELIEF SPACE KNOWLEDGE IN CULTURAL ALGORITHMS

- Domain-Specific Knowledge:
- - Incorporates heuristics (e.g., nearest-neighbor in TSP).
- - Guides variation or repair functions.

INTEGRATING BELIEF SPACE KNOWLEDGE IN CULTURAL ALGORITHMS

- Temporal Knowledge:
- - Tracks fitness trends over generations.
- - Triggers changes in selection pressure or mutation rates.

INTEGRATING BELIEF SPACE KNOWLEDGE IN CULTURAL ALGORITHMS

- Spatial Knowledge:
- - Identifies promising solution regions.
- - Informs sampling, clustering, or adaptive focus for exploration.

INTEGRATING BELIEF SPACE KNOWLEDGE IN CULTURAL ALGORITHMS

- Summary:
- Each knowledge type in the belief space enables strategic influence over population evolution, improving convergence, adaptability, and diversity in cultural algorithms.

INFLUENCE MECHANISMS

- Parameter Adjustment: align with norms.
- Behavioral Modification: encourage/discourage actions.
- Guided Variation: use domain heuristics.
- Trend Analysis: adapt to evolution history.
- Spatial Awareness: focus on strong regions.

BENEFITS OF INFLUENCE FUNCTION

- Accelerated convergence.
- Improved solution quality.
- Preserves diversity.
- Enhances adaptability.

CA CASE STUDY

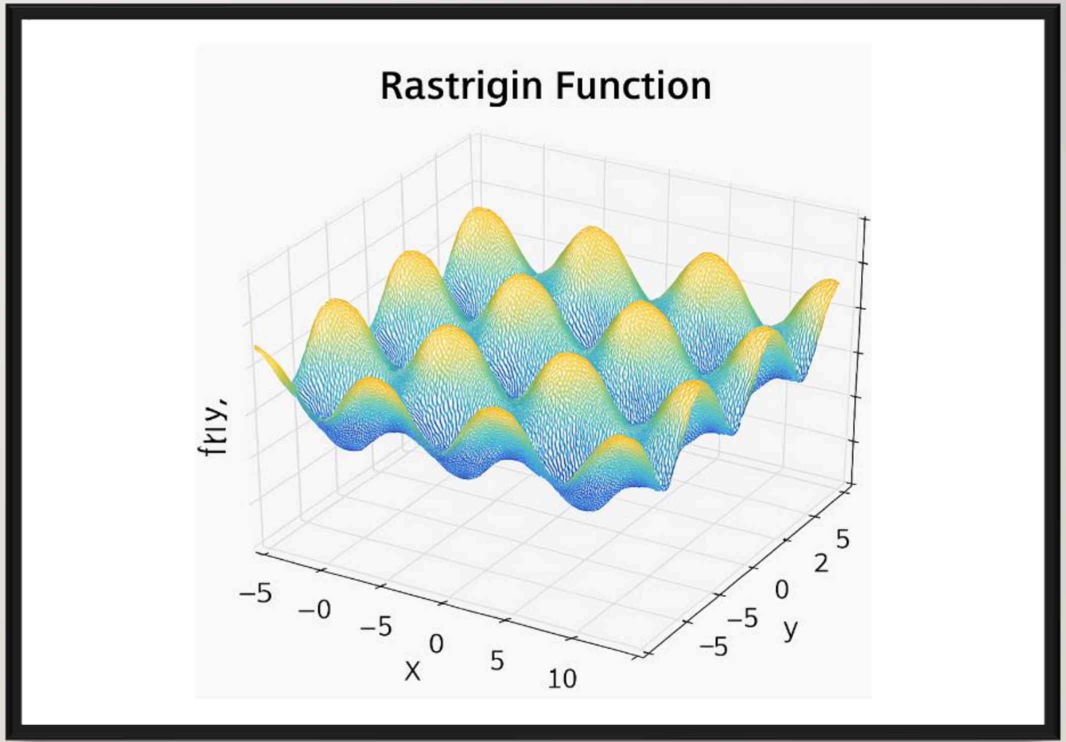
Finding function global optima

PROBLEM DESCRIPTION

- Objective: Minimize the Rastrigin function.
- Multi-dimensional, non-linear, multi-modal.
- Challenges: Many local minima.

THE RASTRIGIN FUNCTION

- Formula: $f(x) = 10n + \Sigma(x_i^2 - 10\cos(2\pi x_i))$
- Global minimum: $x_i = 0$
- Domain: $x_i \in [-5.12, 5.12]$



BELIEF SPACE UPDATE EXAMPLE

- Top-k individuals selected by fitness.
- Mean and variance computed.
- Stored in belief space (cultural knowledge).
- Used to influence next generation.

EXAMPLE: APPLYING INFLUENCE

- Modify individual based on belief space.
- 50% influence is applied per gene i.e. *per allele (dimension)*.
- Sample new value from Gaussian(mean, var).
- Clip to problem bounds.
- Return modified individual.
- Promotes traits aligned with success.

GENETIC OPERATORS

- Crossover: Combines parents into offspring.
- Mutation: Introduces variation.
- Ensures exploration while belief space guides exploitation.

GA VS CA: KEY DIFFERENCES

- GA: Genetic only; CA: Genetic + Cultural.
- GA lacks memory; CA uses belief space.
- GA adapts slowly; CA adapts via knowledge.
- CA maintains diversity & avoids stagnation.

SUMMARY

- CA merges EA and cultural learning.
- Belief space directs evolution.
- Effective on complex, dynamic problems.

CA OPERATORS

CA OPERATORS

1. Parenthood Quantity - influence on no. of children (encourage, suppress)
2. Parenthood Efficiency – increases the proportion of children who will adopt ideas of their parents
3. Proselytism - Religious – ideas spreading beyond children
4. Conservatism – encourage/discourage holding ideas for a long time

CA OPERATORS

5. Propaganda – encourage attacking competing ideas
6. Persuasion – slow convincing of most of the population (spread slowly but cognitively)
7. Motivating – ideas that get adopted via promise of benefits
8. Gradual Integration (Persuasion)
9. Incentives for Adoption (Motivating)

INTEGRATING CULTURAL OPERATORS

- Combine GA's cultural strategies with CA's belief-guided evolution.
- Enrich belief space with behavioral and structural traits.
- Improve adaptability and search control.

TSP CONTEXT EXAMPLES

CA Operators for the Traveling Salesperson Problem

PARENTHOOD QUANTITY

Let's say the belief space has learned that:

"Visiting city, A → B → C is part of most top-performing routes."

Now we:

- Calculate for each individual the **alignment score** = number of belief-endorsed edges present (e.g., A→B, B→C).
- **Bias selection**: Individuals with higher alignment scores get more mating opportunities.
  - Example: If Individual X includes A→B→C, its **reproductive weight** is increased.
  - Individuals lacking these edges are **less likely** to be selected.

PARENTHOOD EFFICIENCY

- After crossover, count how many offspring retained key edges like A→B.
- Update belief space by increasing weights for frequently retained edges.
- Traits with low retention are downgraded.

PROSELYTISM & SHARED MEMORY

- **Proselytism**: Spread elite path segments (e.g., subsequences like [A-B-C]) to the general population.
  - Update belief space with edge sequences common to top-k shortest tours.
  - Influence crossover or mutation operators to preserve or reintroduce these subsequences.
- **Shared Memory**:
  - Store frequently recurring sub-paths (e.g., clusters of cities forming efficient local routes).
  - During offspring generation, prioritize assembling tours from stored high-fitness sub-paths.

CONSERVATISM & ELITISM

- **Conservatism**: Preserve elite tours by reducing disruptive variation.
  - Apply lower mutation rates to elite individuals (shortest tours).
  - Restrict crossover modifications on high-fitness city sequences (e.g., [A-B-C-D]).
- **Elitism**: Directly pass best tours to next generation and belief space.
  - Update belief space using frequent edges from top performers.
  - Ensure elite individuals persist to influence sampling and search direction.

PROPAGANDA & PERSUASION

- **Propaganda**: Penalize city sequences that consistently yield poor tours.
  - Identify recurring edges in low-fitness routes (e.g., [A → F → K]).
  - Reduce influence weight of those edges in belief space.
  - Decrease likelihood of sampling such edges in new offspring.
- **Persuasion**: Reinforce high-performing subsequences gently over generations.
  - Gradually bias influence function toward dominant edge combinations (e.g., [C → D → E]).
  - Allows convergence without sudden loss of diversity.

NEGATIVE SELECTION PRESSURE

- Identify city sequences frequently found in poor-performing tours (e.g., long backtracks or detours like [D → H → A]).
- Penalize those patterns in the belief space by reducing their sampling probability during influence.
- Encourages elimination of non-optimal subpaths across generations.

GRADUAL INTEGRATION (PERSUASION)

- When a subpath like [B → E → G] appears consistently in high-fitness tours, softly bias influence toward preserving this pattern.
- Instead of hard enforcement, adjust sampling distributions so that individuals are more *likely* to adopt parts of such subpaths over time.
- Maintains diversity while pushing population toward consensus tour structures.

INCENTIVIZE ROUTES ALIGNED WITH BELIEF SPACE NORMS

- If belief space favors subpaths like [C → F → G], then individuals whose tours include this subpath receive a **fitness bonus**.
- This encourages reinforcement of effective partial routes.

BOOST EXPLORATION WHEN DIVERSITY DROPS

- If too many individuals converge on similar tours (e.g., nearly identical visit sequences), inject variation:
  - Apply **higher mutation rates**.
  - Introduce **belief-based noise** to explore new permutations.
- Prevents premature convergence and enhances search robustness.

THE END