Algorithmes Évolutionnaires (M2 MIAGE IA²)

Andrea G. B. Tettamanzi
Laboratoire I3S – Équipe SPARKS
andrea.tettamanzi@univ-cotedazur.fr

univ-cotedazur.fr

Séance 1
Introduction

Objectifs de cet enseignement

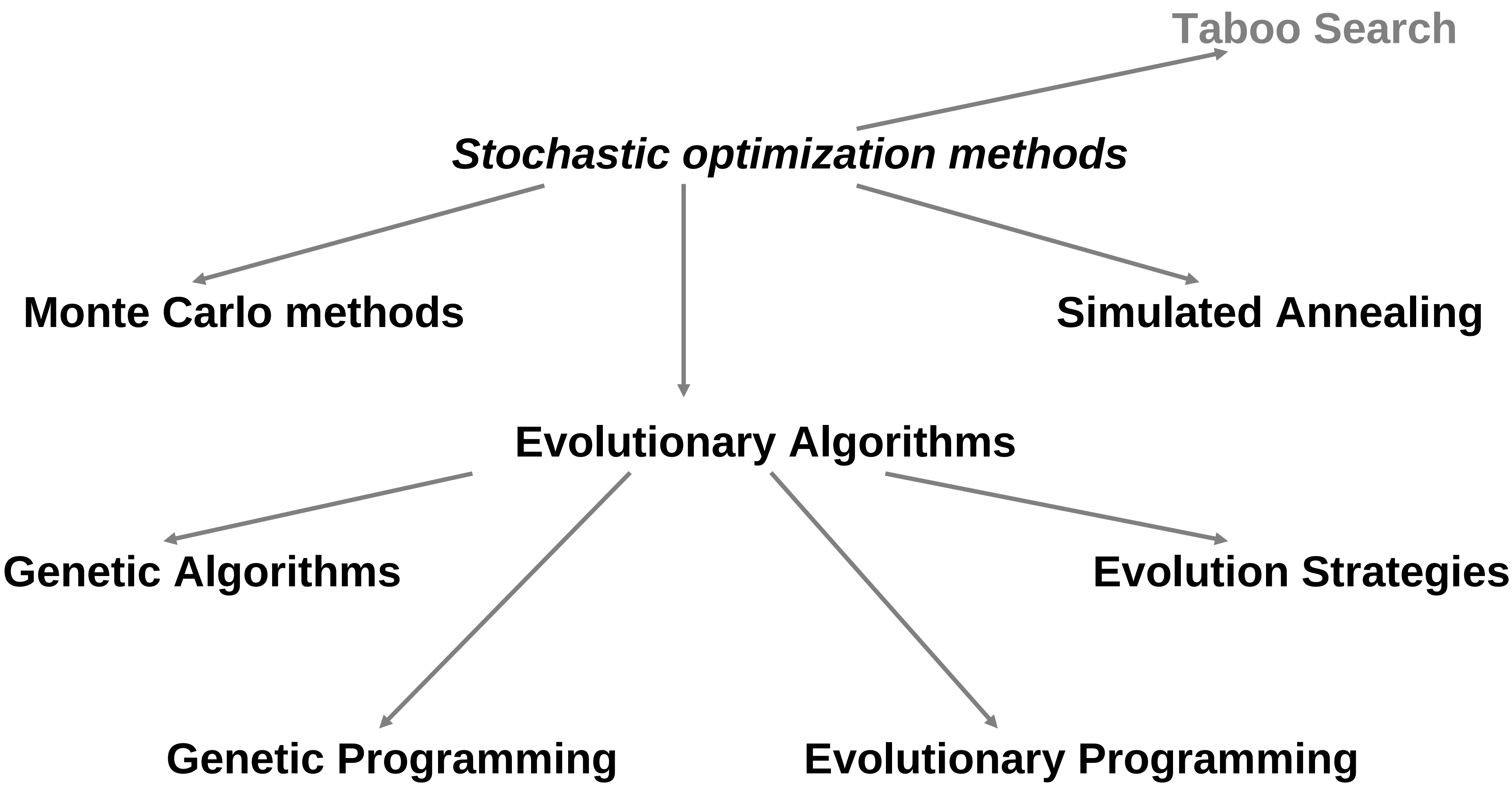
- Fournir une compréhension claire de la métaphore et des concepts sur lesquels repose le calcul évolutionniste
- Fournir une compréhension de certaines des techniques évolutives qui sont devenues des composants essentiels de la boîte à outils de résolution de problèmes du « *soft computing* ».
- Après avoir suivi ce cours, vous saurez
  - ce que l'on entend par algorithme évolutionnaire (AE)
  - comment et pourquoi les AE fonctionnent
  - ce que sont les algorithmes génétiques, la programmation évolutionnaire, les stratégies évolutionnaires et la programmation génétique
  - comment ces techniques peuvent être appliqués à la résolution de problèmes pratiques.

Organisation

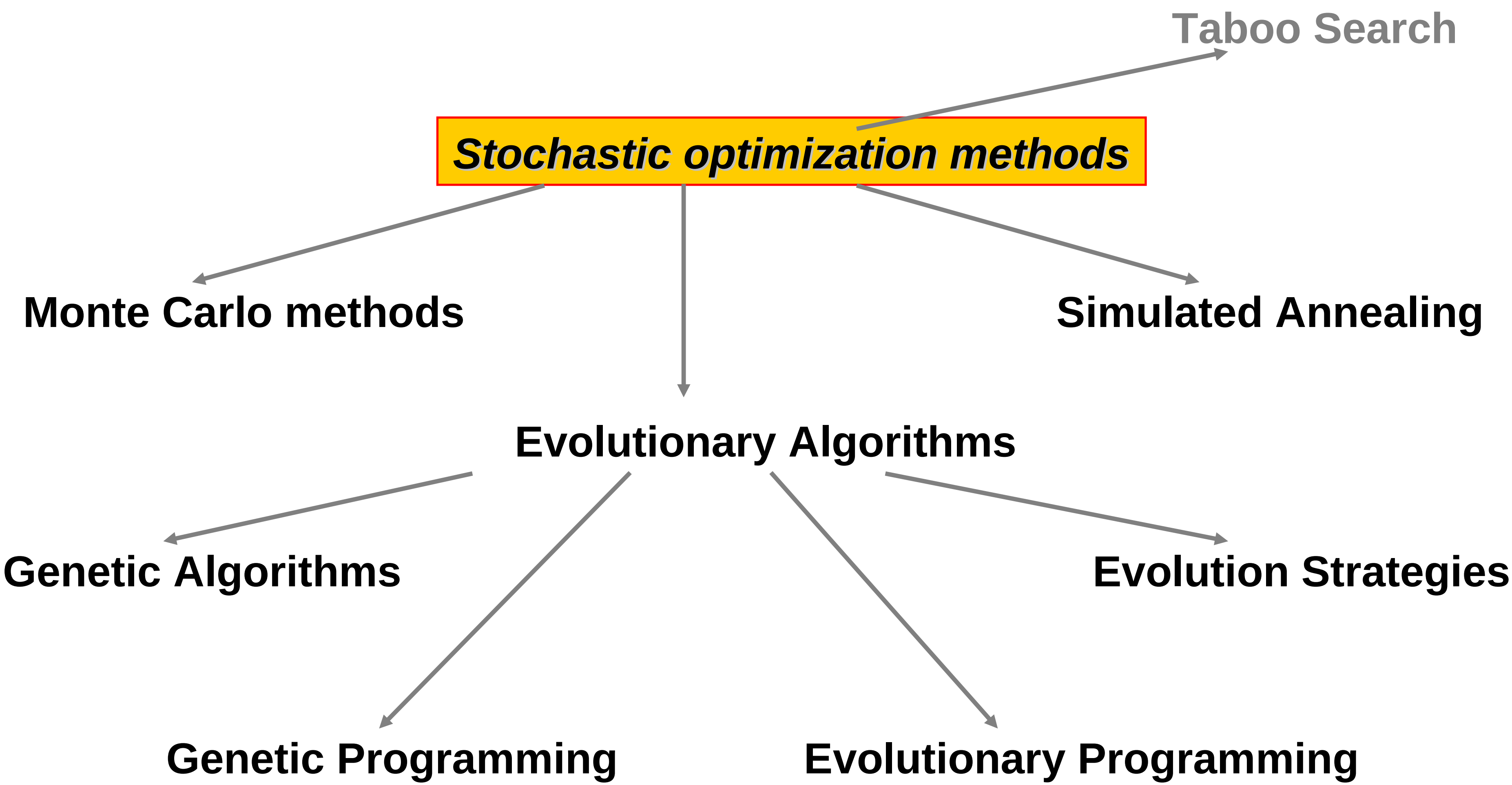
- Page Web (transparents, énoncés, etc.)

  http://www.i3s.unice.fr/~tettaman/Classes/AE/
- Modalités de contrôle des connaissances
  - TP ramassés à chaque séance
  - Contrôle terminal écrit (à confirmer)

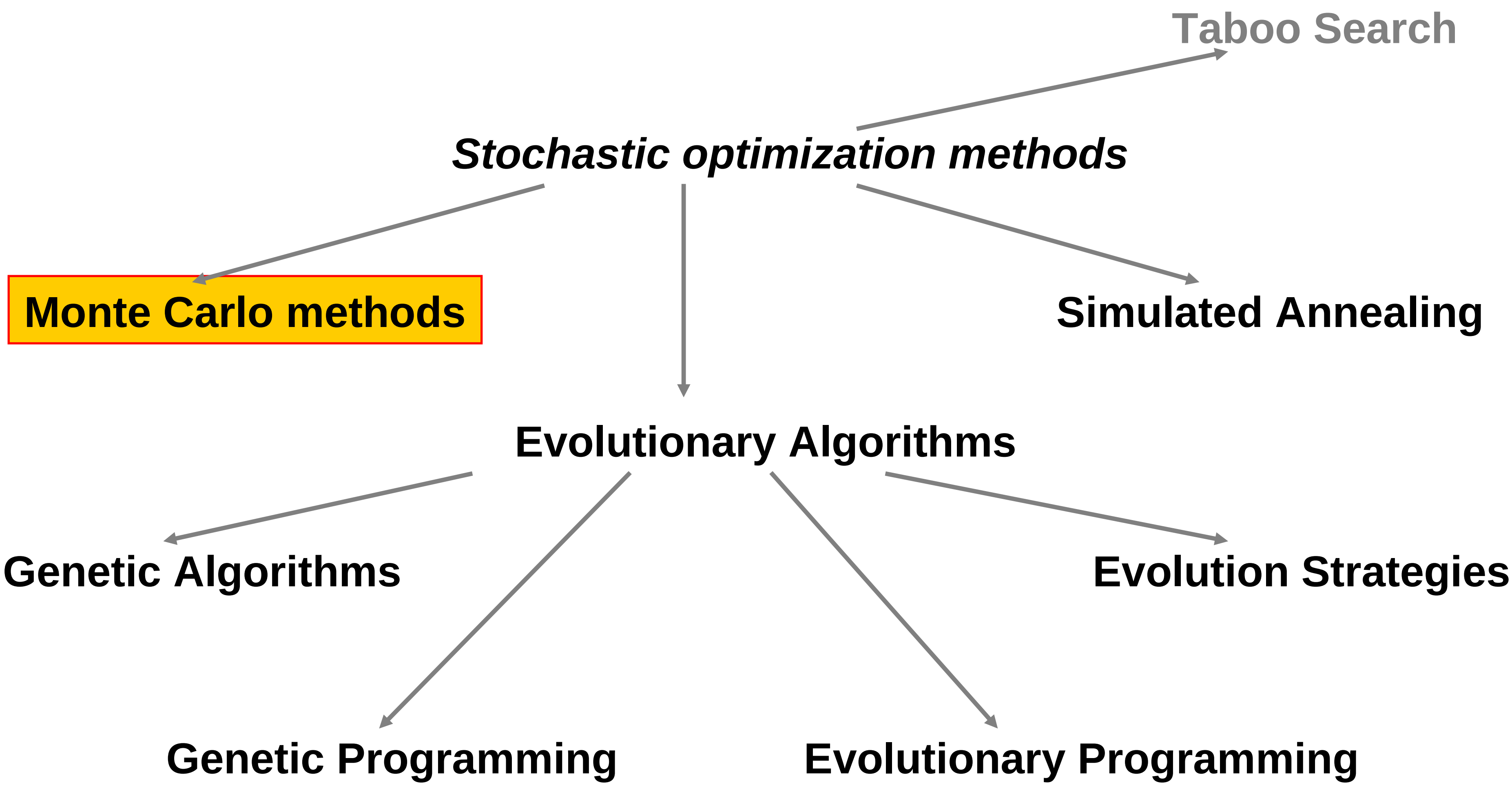
Taxonomy


Taboo Search
Stochastic optimization methods
Monte Carlo methods
Simulated Annealing
Evolutionary Algorithms
Genetic Algorithms
Evolution Strategies
Genetic Programming
Evolutionary Programming

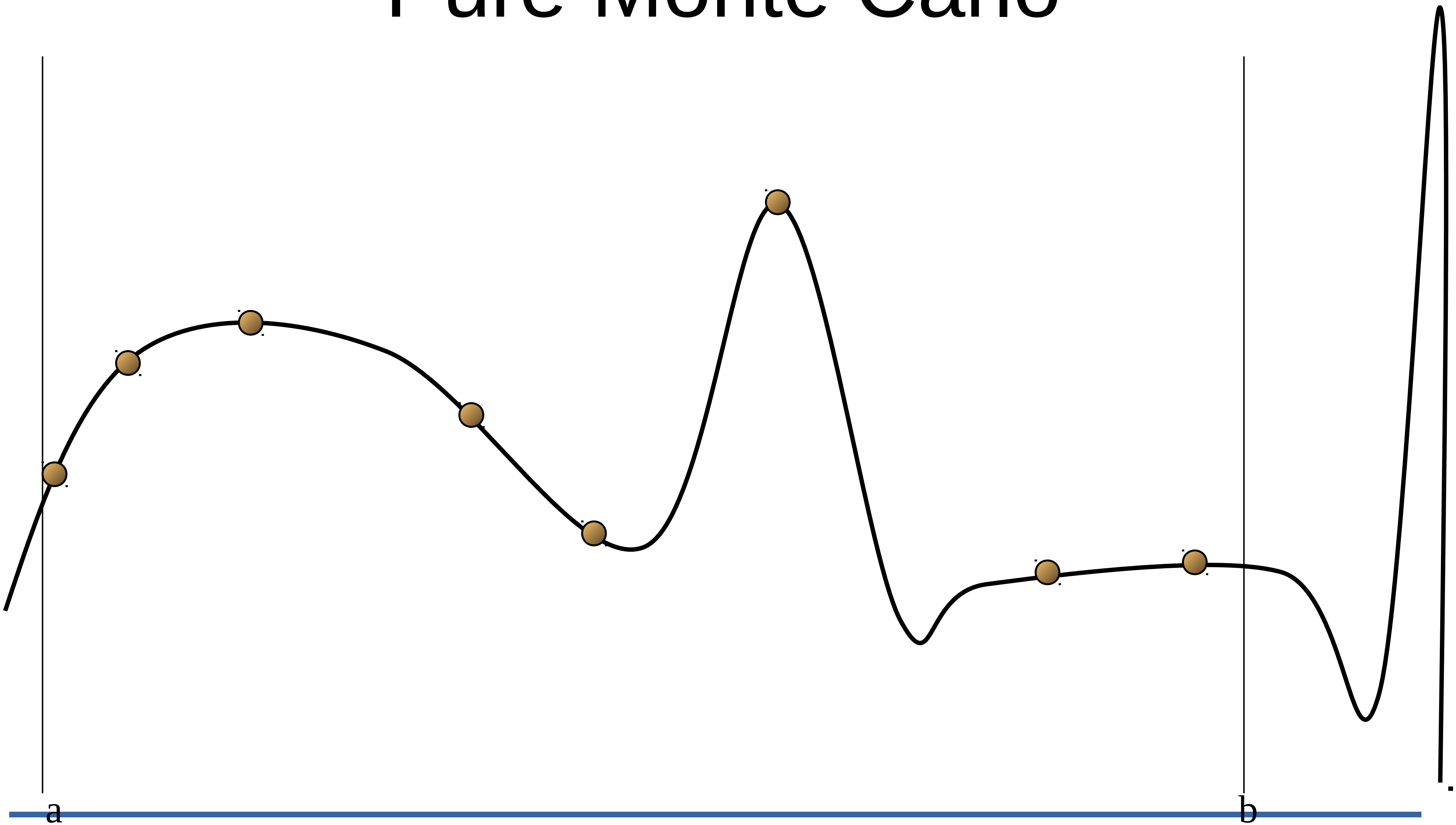
Taxonomy


Taboo Search
Stochastic optimization methods
Monte Carlo methods
Simulated Annealing
Evolutionary Algorithms
Genetic Algorithms
Evolution Strategies
Genetic Programming
Evolutionary Programming

Taxonomy


Taboo Search
Stochastic optimization methods
Monte Carlo methods
Simulated Annealing
Evolutionary Algorithms
Genetic Algorithms
Evolution Strategies
Genetic Programming
Evolutionary Programming

Pure Monte Carlo

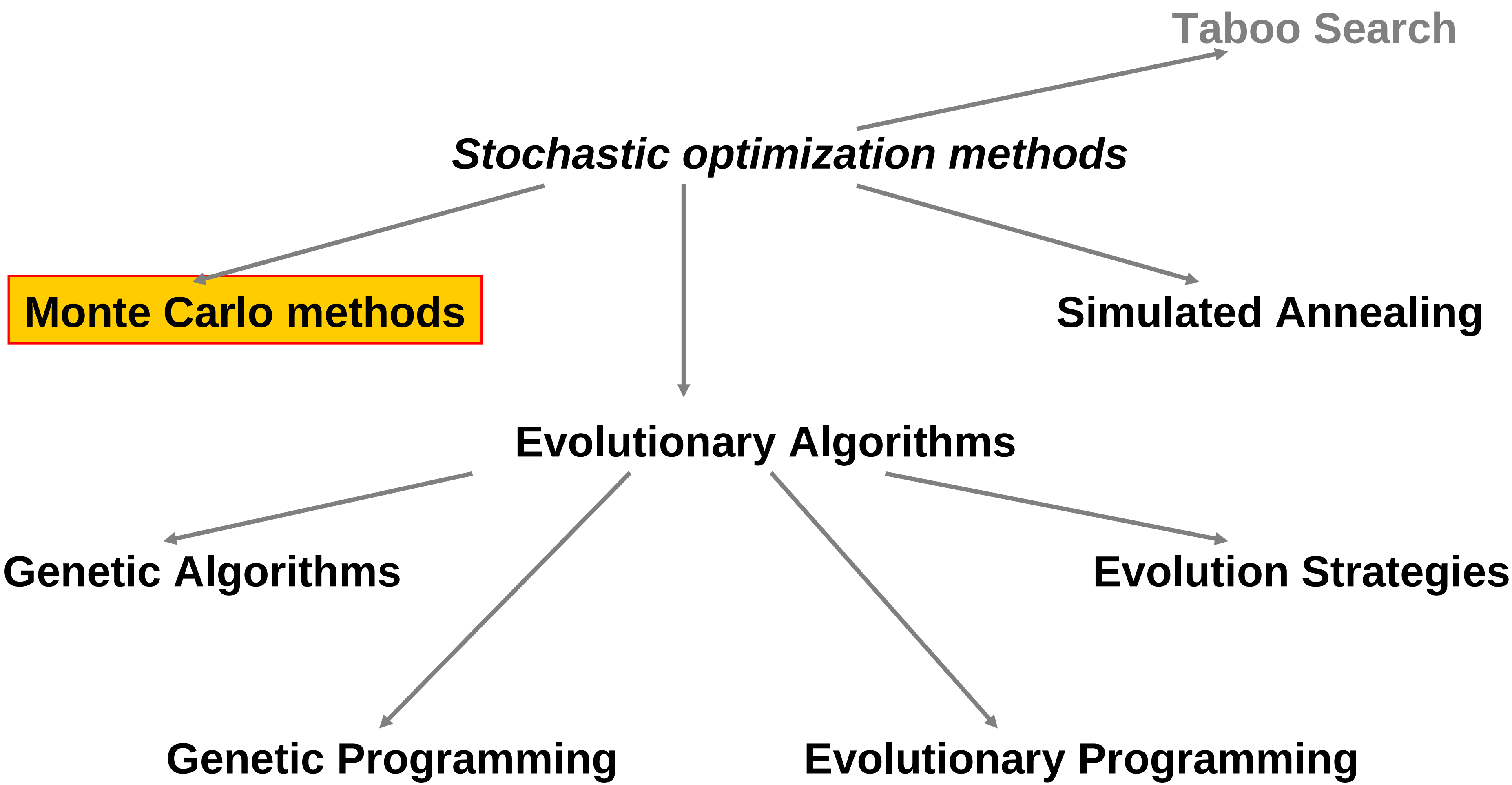


Convergence in probability

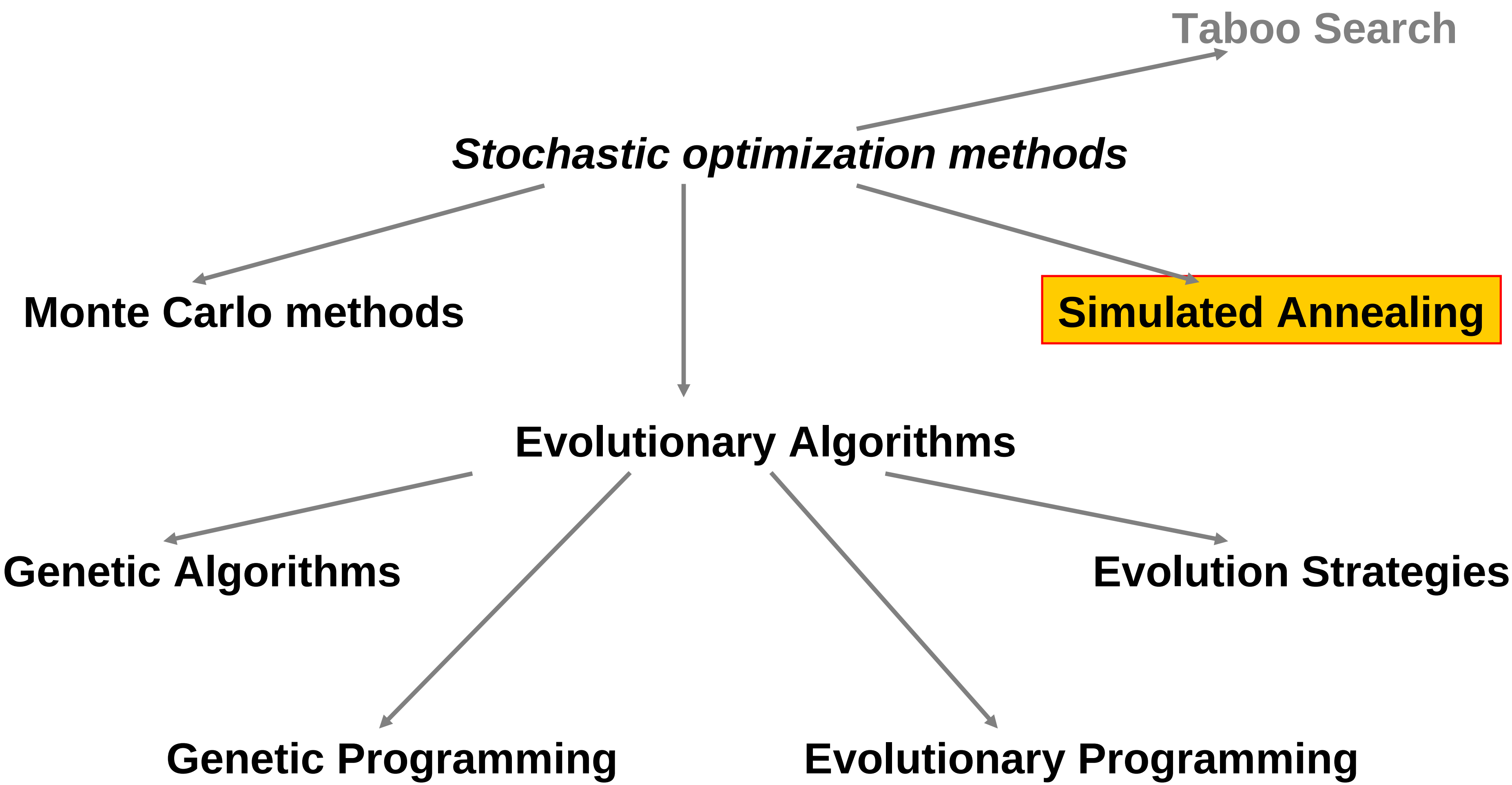
$$\lim_{t \to \infty} \Pr\left[\left| f(x_t) - f(x^*) \right| < \varepsilon\right] = 1$$

$$\lim_{t \to \infty} f(x_t) = f(x^*)$$ **"almost surely"**

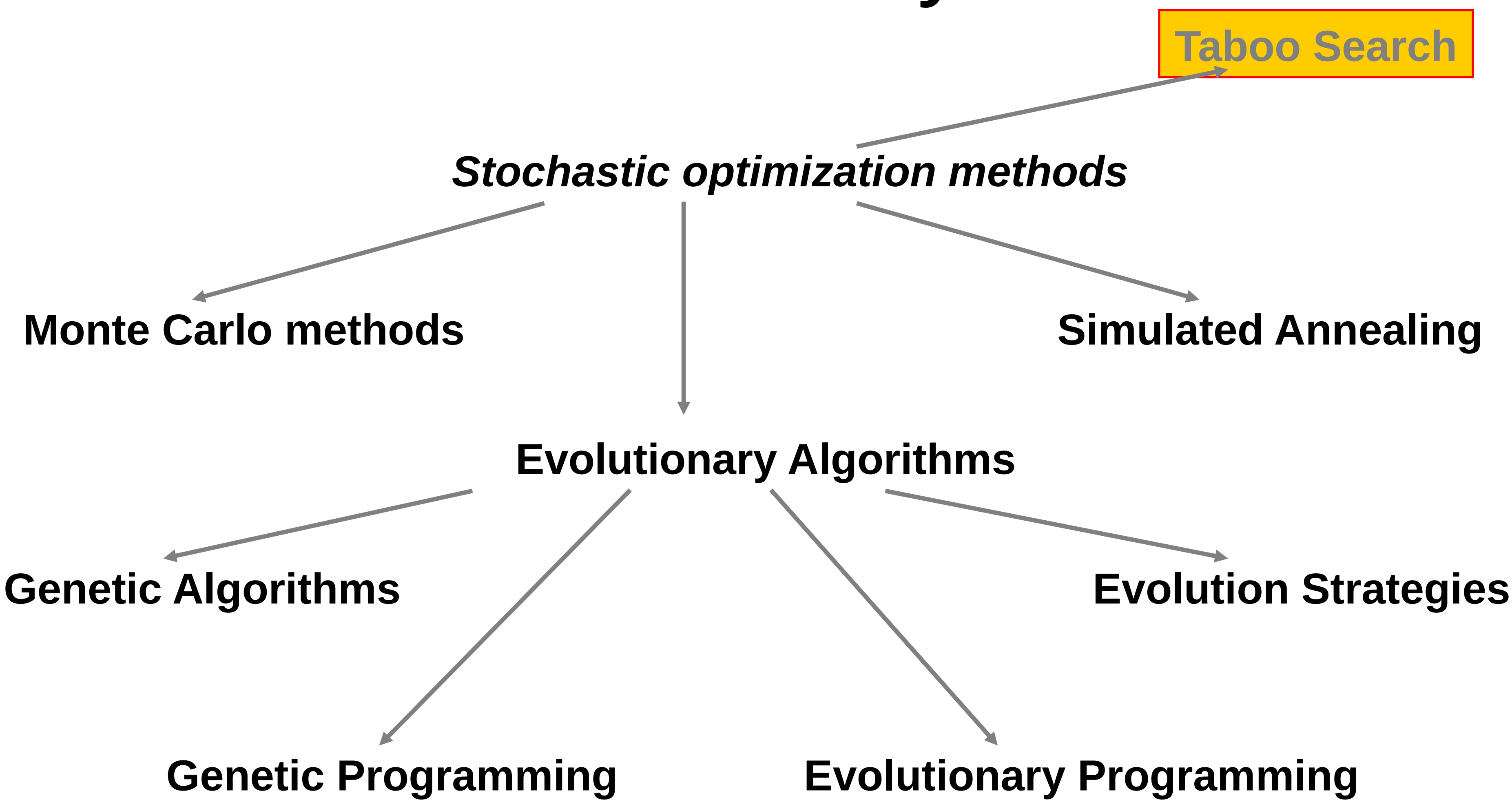
Taxonomy


Taboo Search
Stochastic optimization methods
Monte Carlo methods
Simulated Annealing
Evolutionary Algorithms
Genetic Algorithms
Evolution Strategies
Genetic Programming
Evolutionary Programming

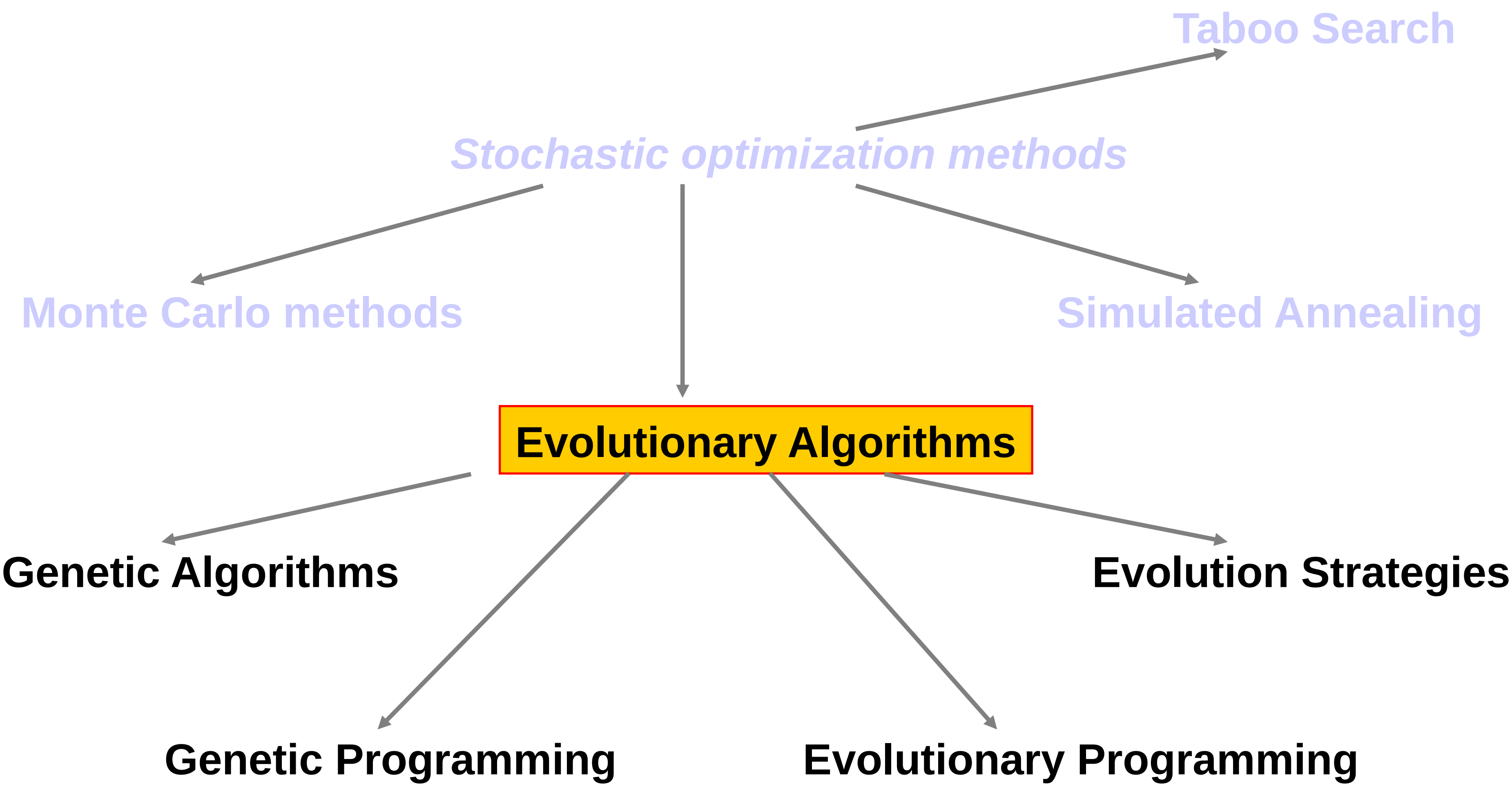
Taxonomy


Taboo Search
Stochastic optimization methods
Monte Carlo methods
Simulated Annealing
Evolutionary Algorithms
Genetic Algorithms
Evolution Strategies
Genetic Programming
Evolutionary Programming

Taxonomy


Taboo Search
Stochastic optimization methods
Monte Carlo methods
Simulated Annealing
Evolutionary Algorithms
Genetic Algorithms
Evolution Strategies
Genetic Programming
Evolutionary Programming

Taxonomy

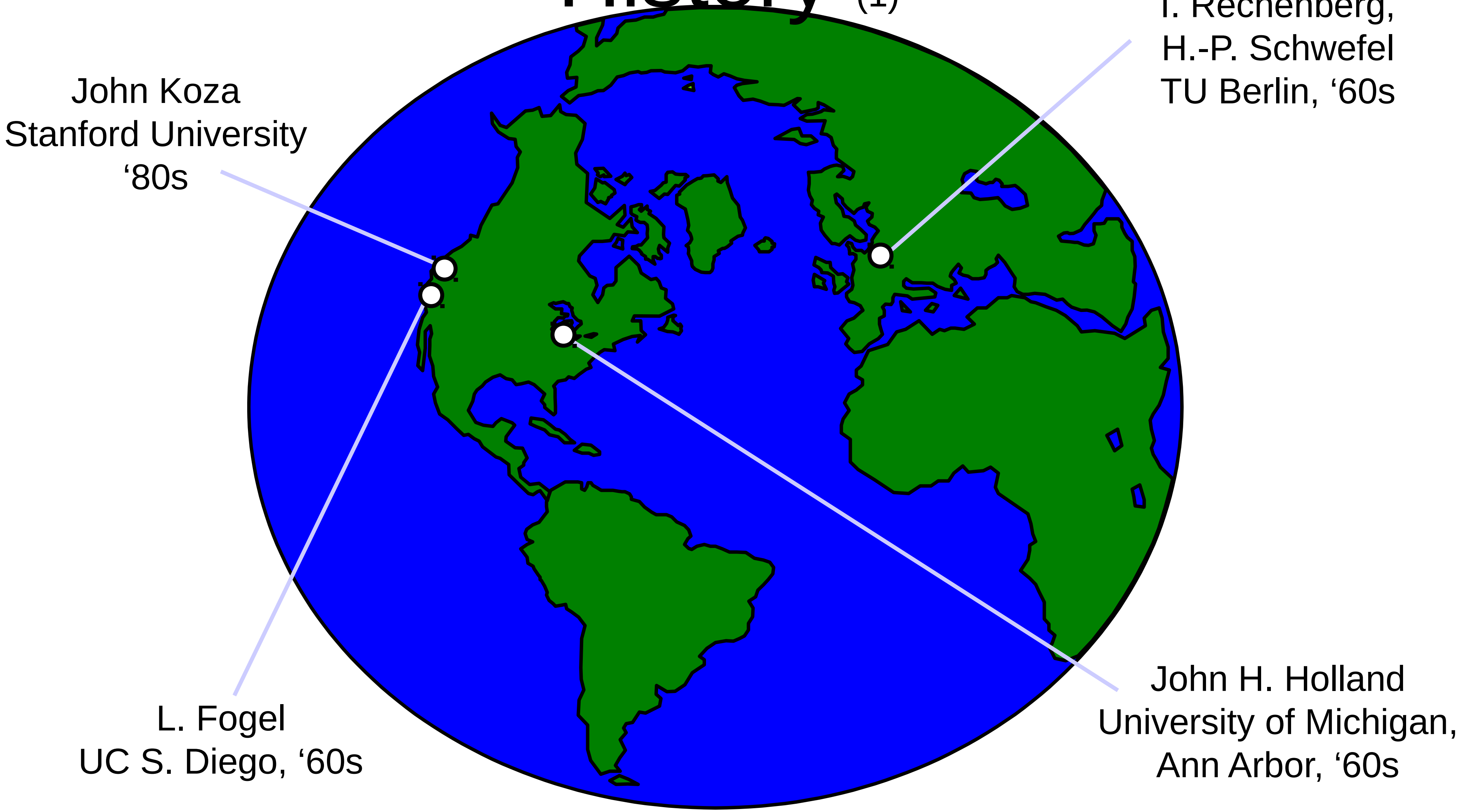


Taboo Search
Stochastic optimization methods
Monte Carlo methods
Simulated Annealing
Evolutionary Algorithms
Genetic Algorithms
Evolution Strategies
Genetic Programming
Evolutionary Programming

Distinctive Features of EAs

- operate on appropriate encoding of solutions;
- population search;
- no regularity conditions requested;
- probabilistic transitions.

History (1)



John Koza
Stanford University
'80s
I. Rechenberg,
H.-P. Schwefel
TU Berlin, '60s
L. Fogel
UC S. Diego, '60s
John H. Holland
University of Michigan,
Ann Arbor, '60s

History (2)

1859 Charles Darwin: inheritance, variation, natural selection

1957 G. E. P. Box: random mutation & selection for optimization

1958 Fraser, Bremermann: computer simulation of evolution

1964 Rechenberg, Schwefel: mutation & selection

1966 Fogel et al.: evolving automata - "evolutionary programming"

History (3)

1975 Holland: crossover, mutation & selection - “reproductive plan”

1975 De Jong: parameter optimization - “genetic algorithm”

1989 Goldberg: first textbook

1991 Davis: first handbook

1992 Michalewicz: Genetic Algorithms + Data Structures = Evolution Programs

History (4)

- 1993 Koza: evolving LISP programs - “genetic programming”
- 1998 Ryan & O’Neill: Grammatical Evolution
- 1999 Storn & Price: Differential Evolution
- 2001 Poli: schema theorem for GP
- 2002 Stanley & Miikkulainen: NEAT

The Metaphor

EVOLUTION	PROBLEM SOLVING
Environment	Object problem
Individual	Candidate solution
Fitness	Quality

The Metaphor

EVOLUTION	PROBLEM SOLVING
Environment	**Object problem**
Individual	Candidate solution
Fitness	Quality

The Metaphor

EVOLUTION	PROBLEM SOLVING
Environment	Object problem
Individual	**Candidate solution**
Fitness	Quality

The Metaphor

EVOLUTION	PROBLEM SOLVING
Environment	Object problem
Individual	Candidate solution
Fitness	**Quality**

Alternative Views of Evolutionary Algorithms

- Operations Research: optimization method
- Decision Theory: optimal decision
- Machine Learning: learning technique
- Artificial Life: artificial counterpart of natural evolution
- Biology: tool for testing evolutionary models

Optimization Problem

Cost function $c : S \to \mathbf{R}$

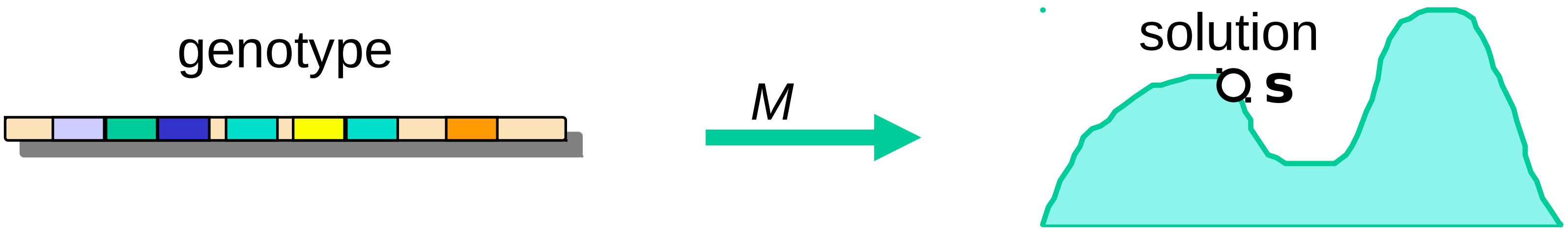
minimize $c(\mathbf{s})$

subject to $\mathbf{s} \in S_{FEAS}.$

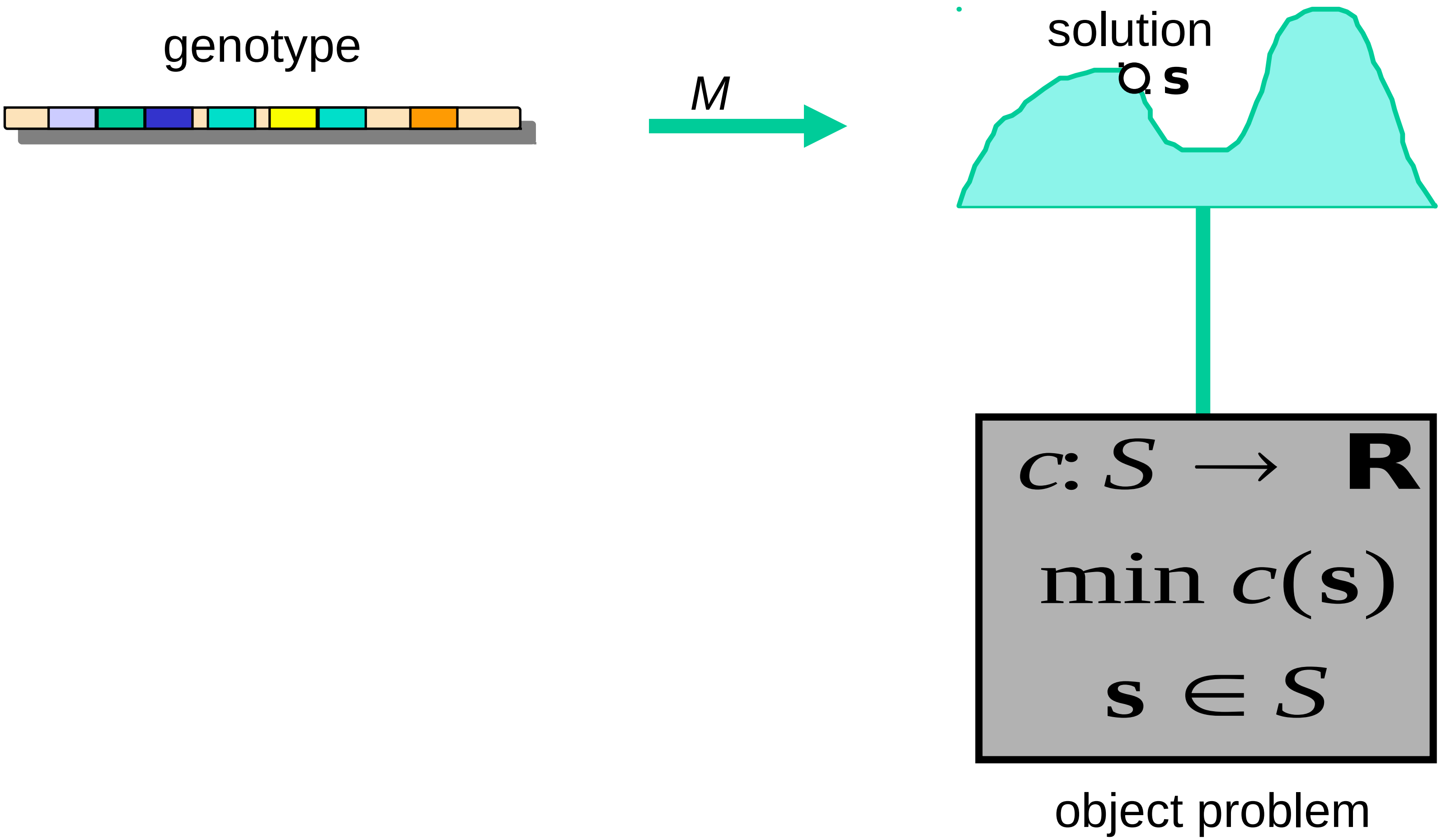
Object problem and Fitness

genotype

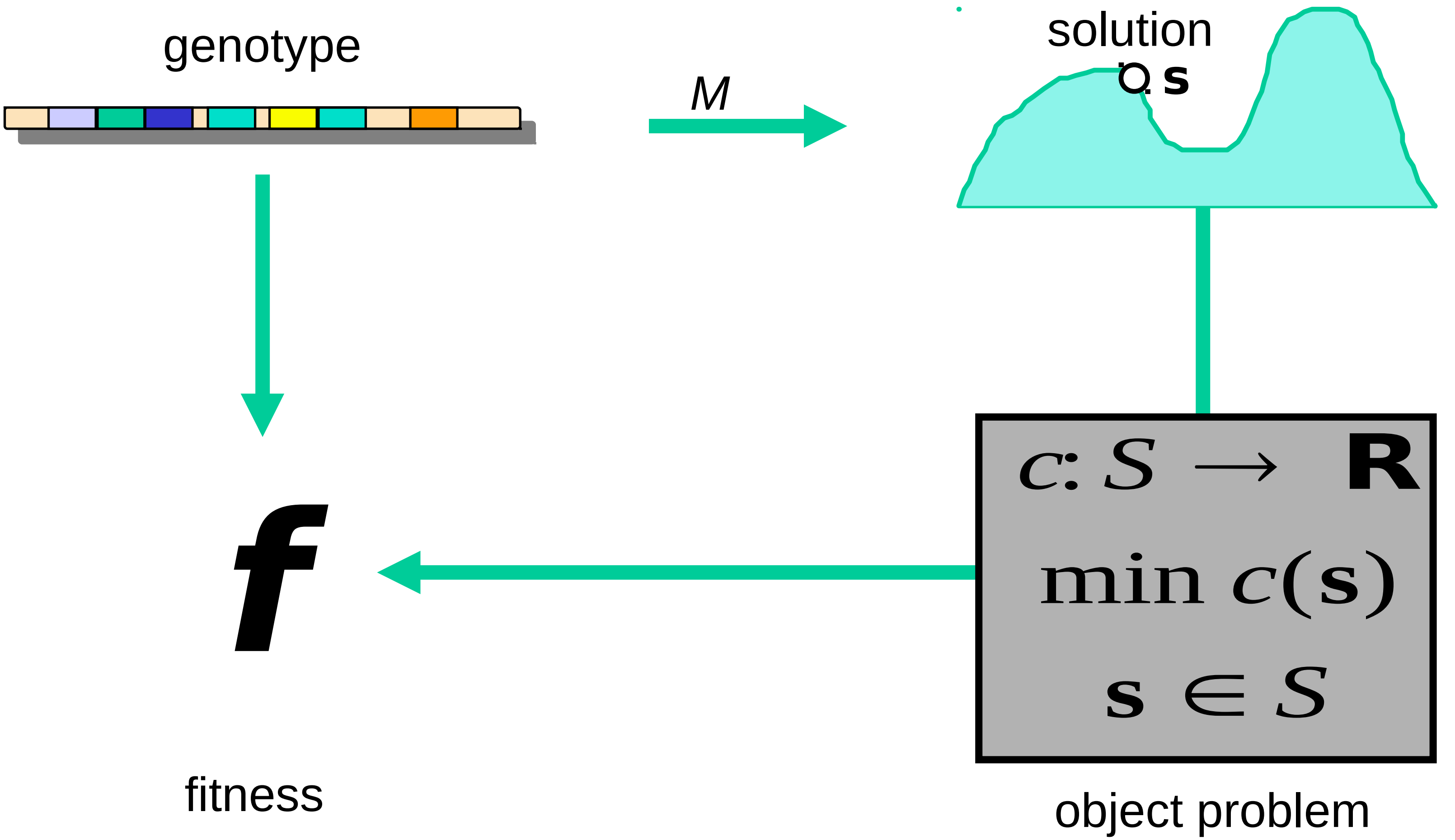
Object problem and Fitness



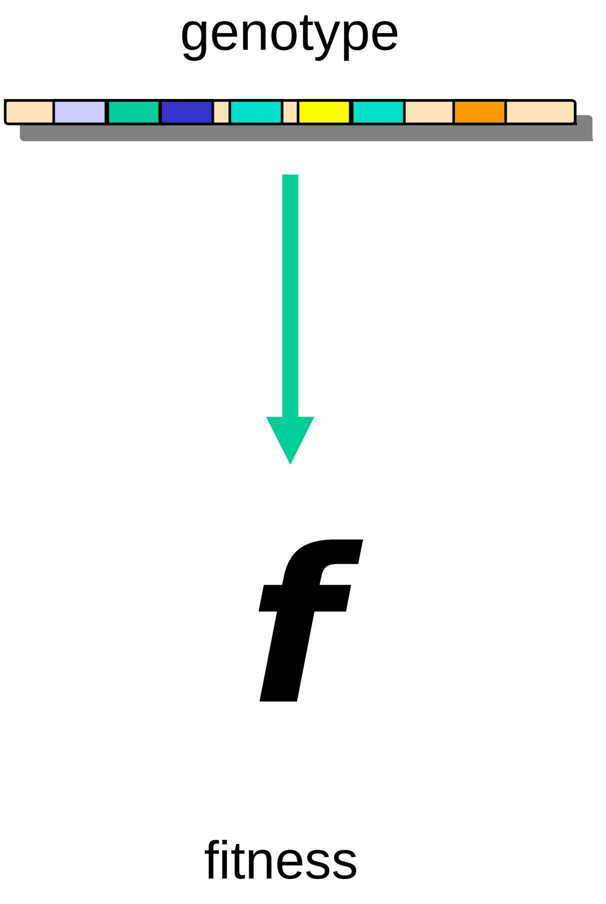
Object problem and Fitness



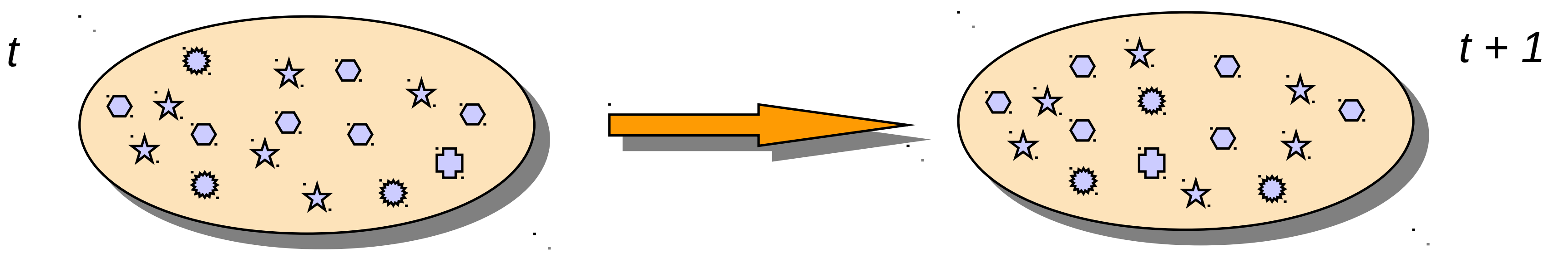
Object problem and Fitness



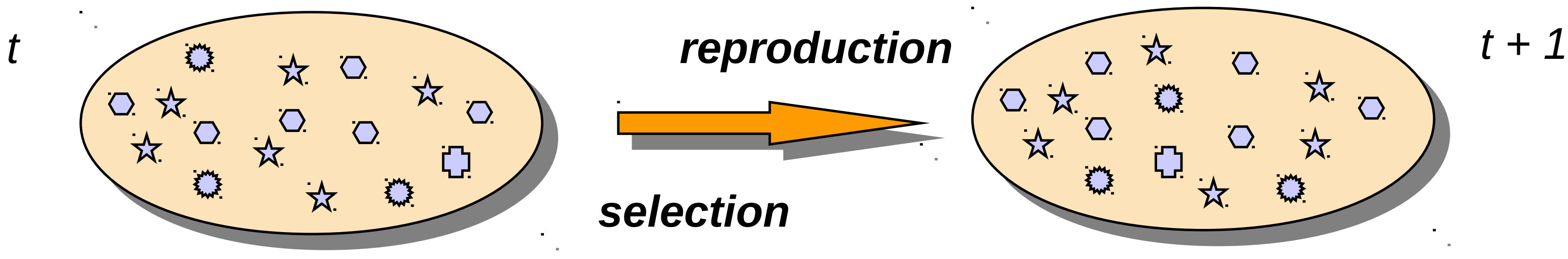
Object problem and Fitness



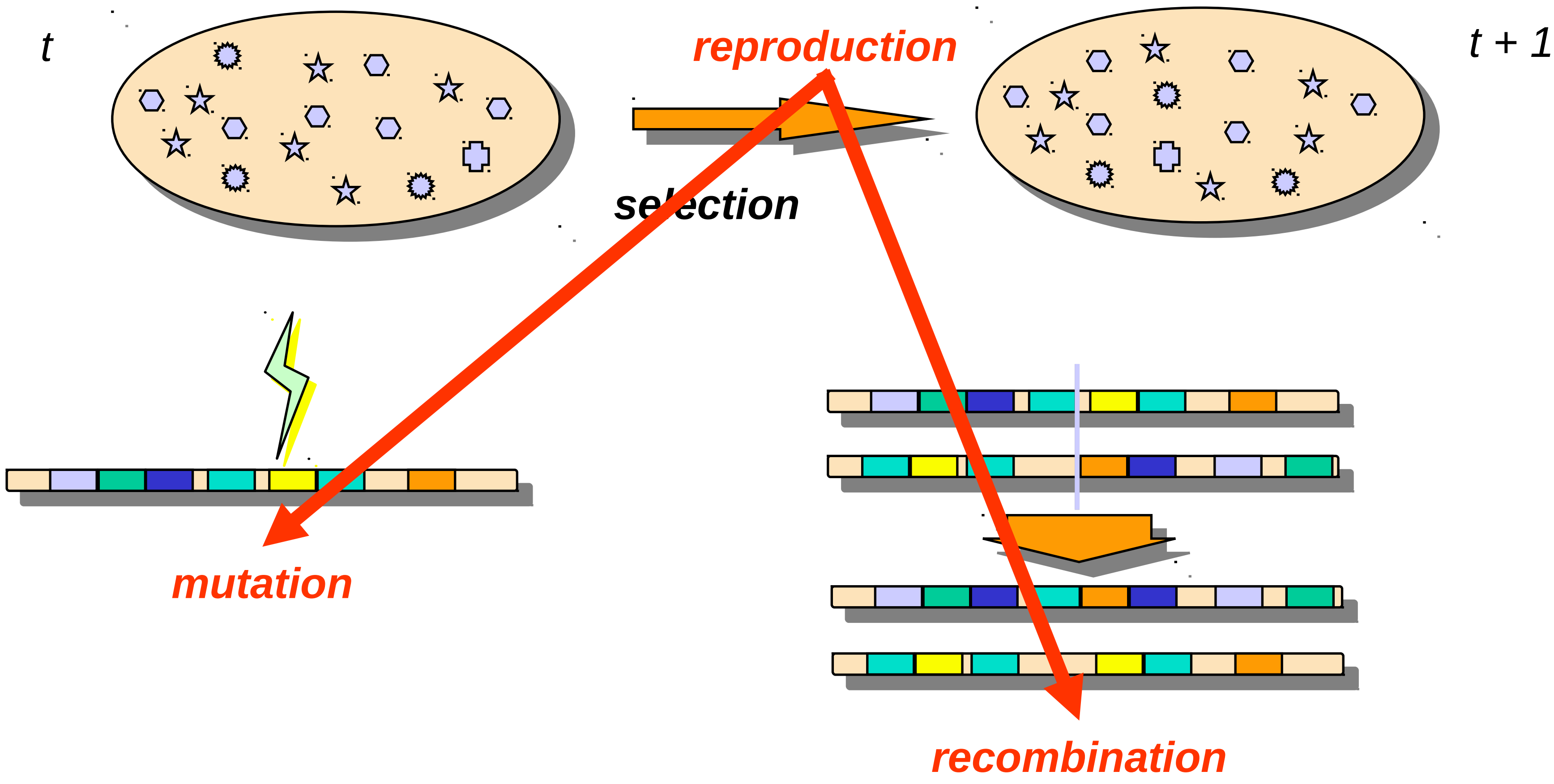
The Ingredients



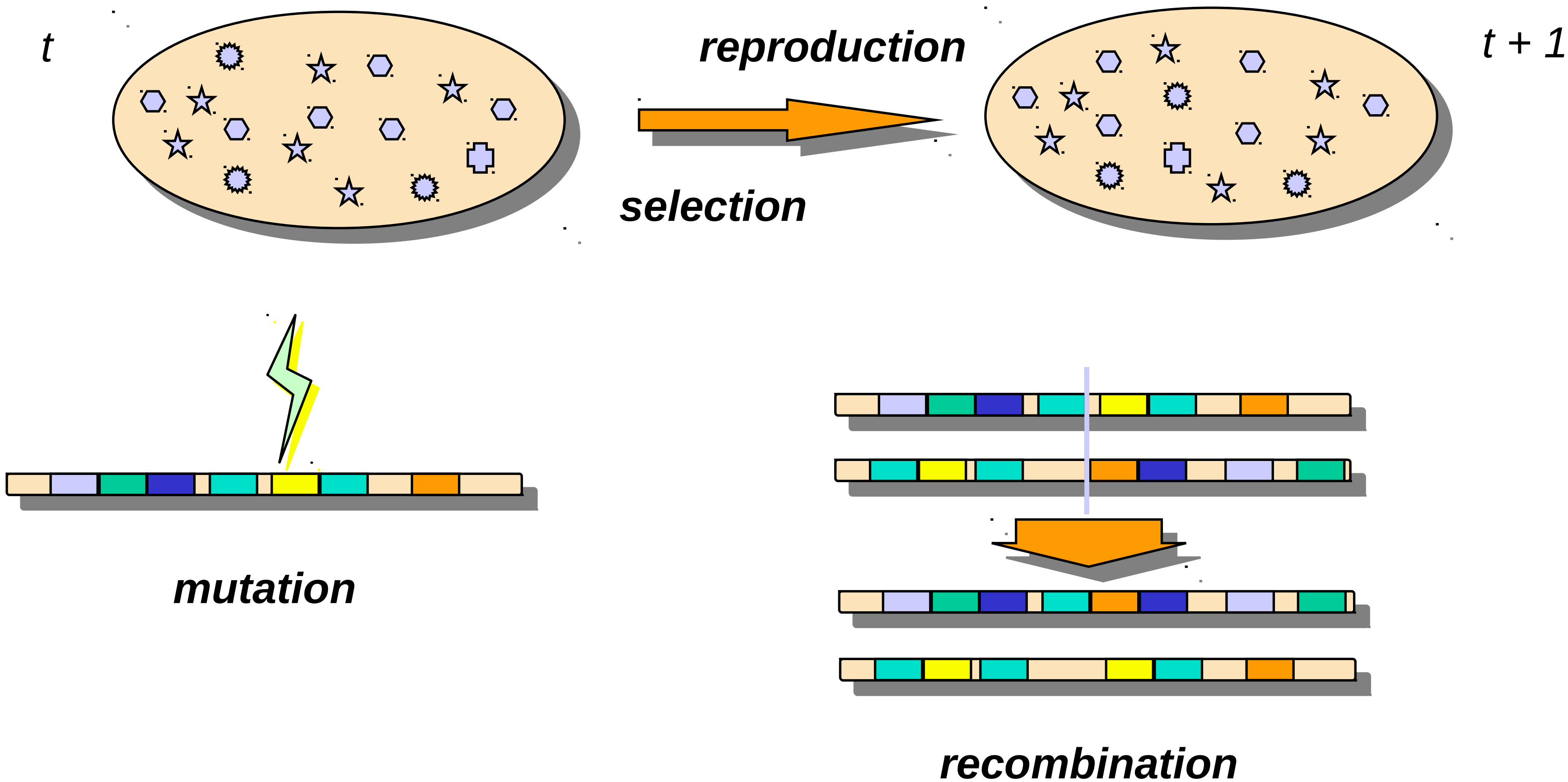
The Ingredients



The Ingredients



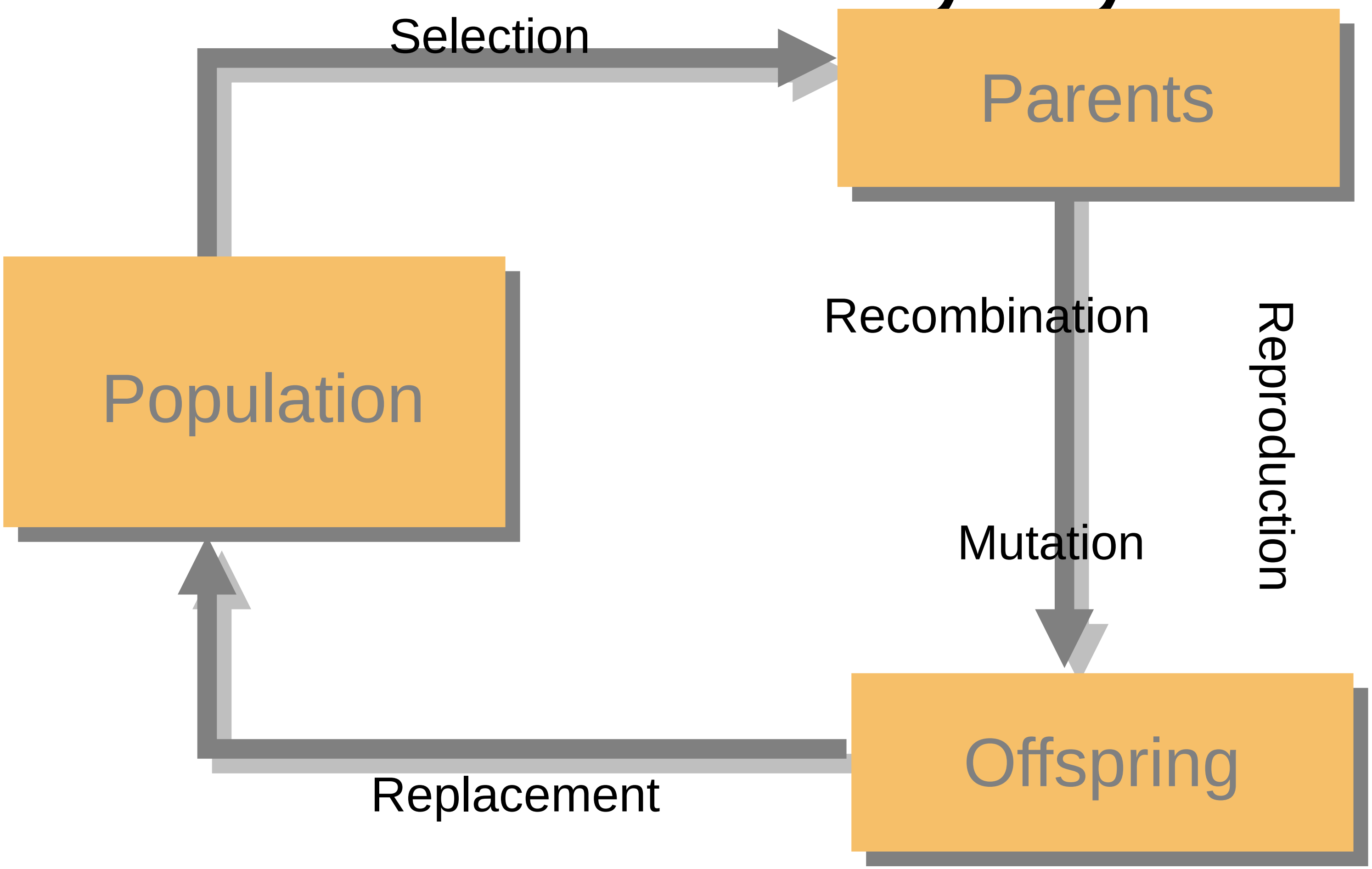
The Ingredients



Pseudocode

```
generation = 0;
SeedPopulation(popSize); // at random or from a file
while(!TerminationCondition())
{
    generation = generation + 1;
    CalculateFitness();        // ... of new genotypes
    Selection();               // select genotypes that will reproduce
    Crossover(p_cross);        // mate p_cross of them on average
    Mutation(p_mut);           // mutate all the offspring with Bernoulli
                               // probability p_mut over genes
}
```

The Evolutionary Cycle

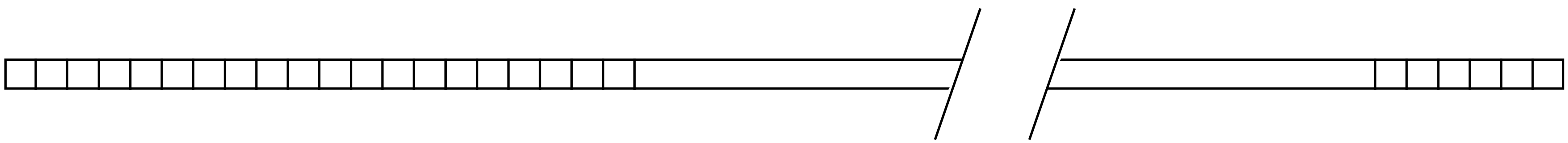


A Sample Genetic Algorithm

- The MAXONE problem
- Genotypes are bit strings
- Fitness-proportionate selection
- One-point crossover
- Flip mutation (transcription error)

The MAXONE Problem

Problem instance: a string of *l* binary cells, $\gamma \in \{0, 1\}^l$:



Fitness:

$$f(\gamma) = \sum_{i=1}^{l} \gamma_i$$

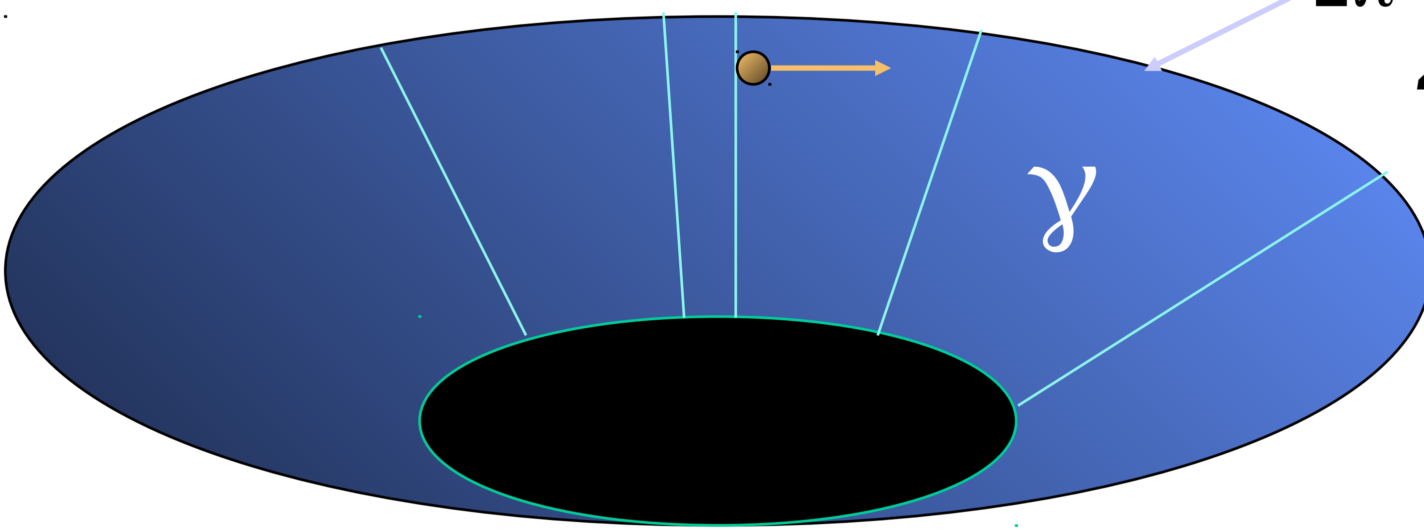
Objective: maximize the number of ones in the string.

Fitness Proportionate Selection

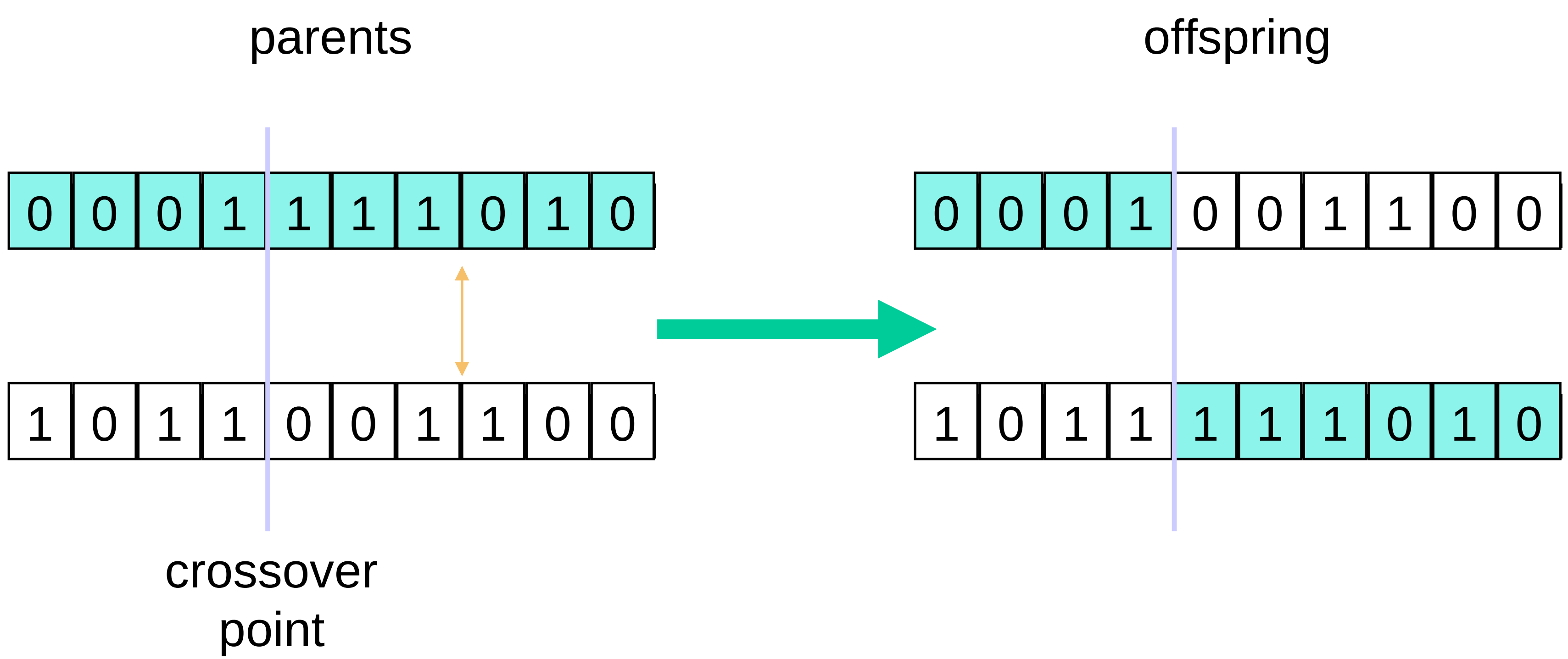
Probability of γ being selected: $$P(\gamma) = \frac{f(\gamma)}{\sum f}$$

Implementation: “Roulette Wheel”

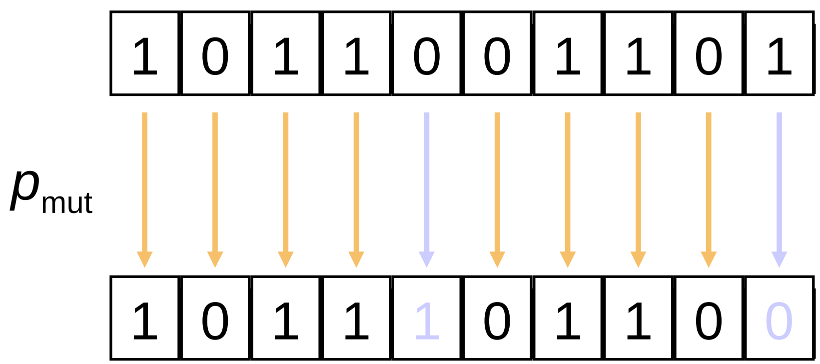
$$2\pi \frac{f(\gamma)}{\sum f}$$



One-Point Crossover



Mutation



independent Bernoulli transcription errors

Example: Selection

0111011011	f = 7	Cf = 7	P = 0.125
1011011101	f = 7	Cf = 14	P = 0.125
1101100010	f = 5	Cf = 19	P = 0.089
0100101100	f = 4	Cf = 23	P = 0.071
1100110011	f = 6	Cf = 29	P = 0.107
1111001000	f = 5	Cf = 34	P = 0.089
0110001010	f = 4	Cf = 38	P = 0.071
1101011011	f = 7	Cf = 45	P = 0.125
0110110000	f = 4	Cf = 49	P = 0.071
0011111101	f = 7	Cf = 56	P = 0.125

Random sequence: 43, 1, 19, 35, 15, 22, 24, 38, 44, 2

Example: Recombination & Mutation

0111011011	→	0111011011	→	0111111011	f = 8
0111011011	→	0111011011	→	0111011011	f = 7
110 | 1100010	→	1100101100	→	1100101100	f = 5
010 | 0101100	→	0101100010	→	0101100010	f = 4
1 | 100110011	→	1100110011	→	1100110011	f = 6
1 | 100110011	→	1100110011	→	1000110011	f = 5
0110001010	→	0110001010	→	0110001010	f = 4
1101011011	→	1101011011	→	1101011011	f = 7
011000 | 1010	→	0110001011	→	0110001011	f = 5
110101 | 1011	→	1101011010	→	1101011010	f = 6

Example: Replacement

0111011011	f = 7	0111111011	f = 8
1011011101	f = 7	0111011011	f = 7
1101100010	f = 5	1100101100	f = 5
0100101100	f = 4	0101100010	f = 4
1100110011	f = 6	1100110011	f = 6
1111001000	f = 5	1000110011	f = 5
0110001010	f = 4	0110001010	f = 4
1101011011	f = 7	1101011011	f = 7
0110110000	f = 4	0110001011	f = 5
0011111101	f = 7	1101011010	f = 6
TOTAL = 56		TOTAL = 57