A Genetic Quantum Annealing Algorithm

Luca Armando Nutricati

Institute for Particle Physics Phenomenology,
Durham University

28/10/2022

Motivation

- **Genetic algorithms (GA)** are a valid tools to tackle **search** and **optimisation problems** *(from sudoku puzzles to string theory landscapes…)*

- However, for some problems, the **search space can be very large** *(e.g. string theory landscapes ~ 10^{500})*

- Classical genetic algorithms may **not be efficient**

Motivation

- **Genetic algorithms (GA)** are a valid tools to tackle **search** and **optimisation problems** *(from sudoku puzzles to string theory landscapes…)*

- However, for some problems, the **search space can be very large** *(e.g. string theory landscapes ~ 10^{500})*

- Classical genetic algorithms may **not be efficient**

Can we construct an enhanced version of genetic algorithms using quantum computing?

Outline

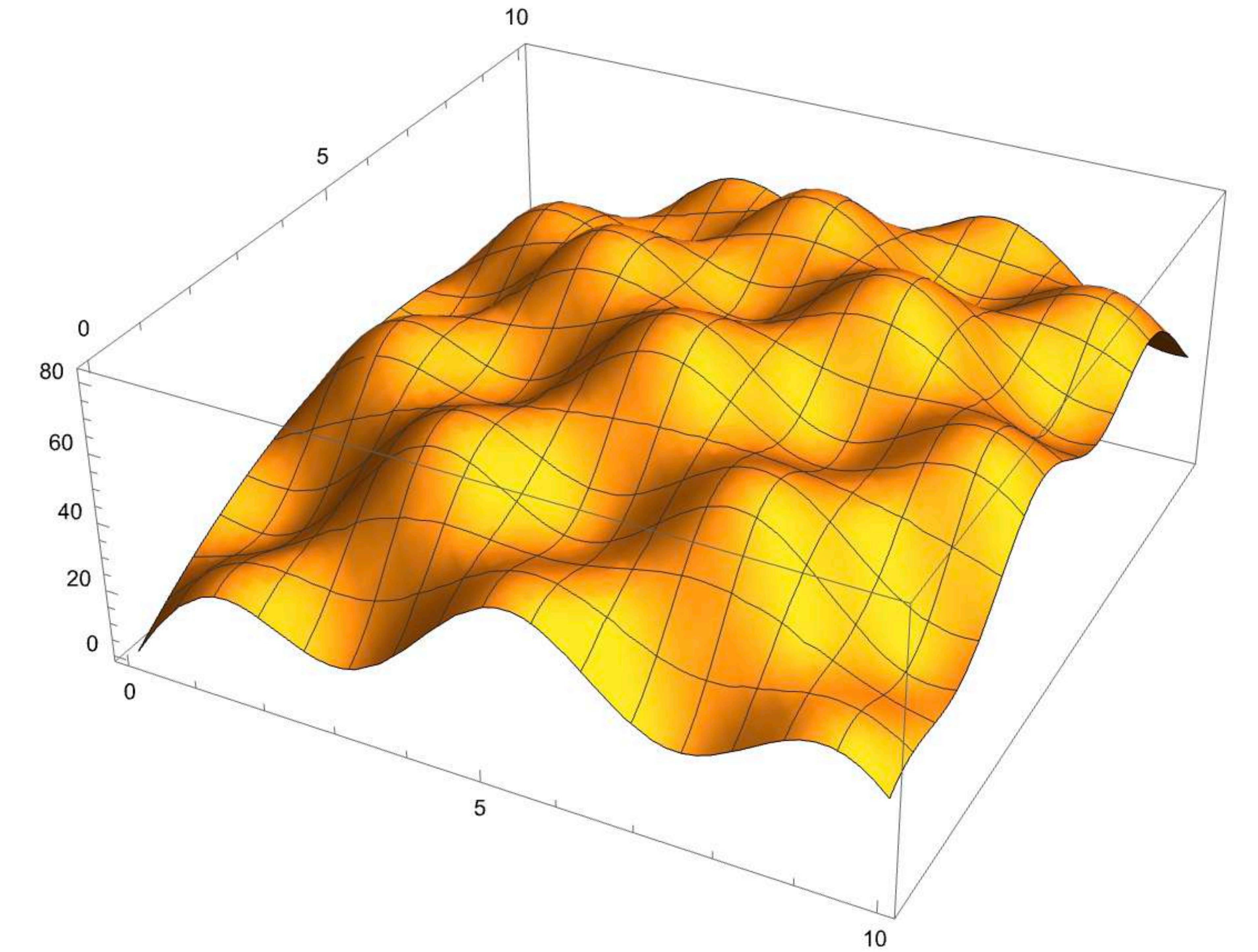
- **Background on Genetic Algorithms (GA)**
- **Introduction to Quantum Annealing**
- **The combined technique: Genetic Quantum Annealing (GQA)**
- **GA vs GQA**

Background on Genetic Algorithms

A **genetic algorithm** (**GA**) is a **heuristic search algorithm** inspired by the process of natural selection.

Genetic algorithms are used to generate high-quality solutions to **optimisation and search problems** by relying on biologically inspired operators such as **mutation**, **crossover** and **selection**.

Example: find global maximum to 250 decimal places without using calculus

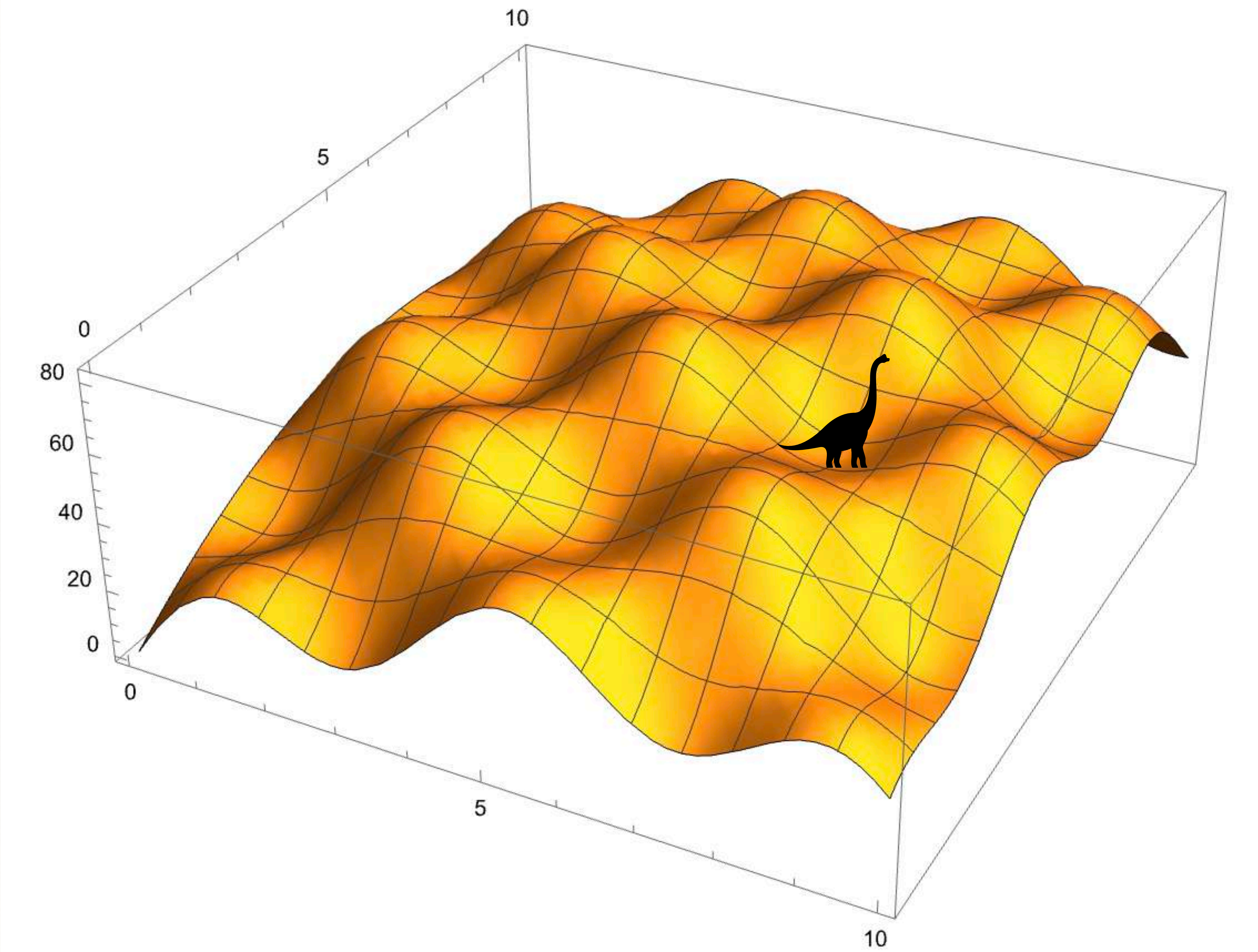


$$f(x,y) = 12\left(\cos\frac{3y}{2}\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

Search space: 10^{500}

Background on Genetic Algorithms

Example: find global maximum to 250 decimal places without using calculus



$$f(x,y) = 12\left(\cos\frac{3y}{2}\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

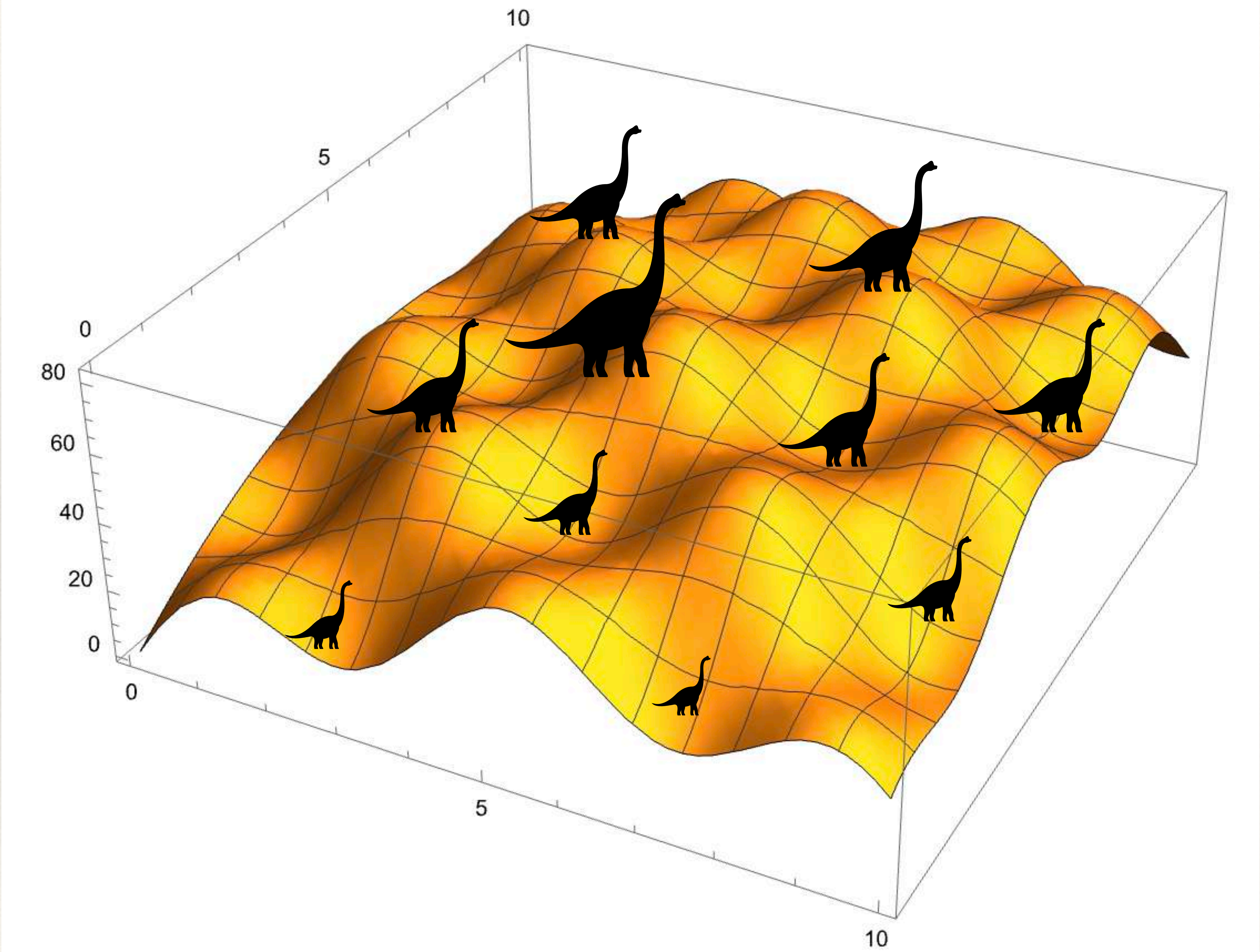
Define a creature and its genotype:

Genotype ➡ *(x,y)*   *x = a.bcdef …*
y = g.hijkl …

Phenotype ➡ *f(x,y)*

Background on Genetic Algorithms

Example: find global maximum to 250 decimal places without using calculus



$$f(x, y) = 12\left(\cos\frac{3y}{2}\ \sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

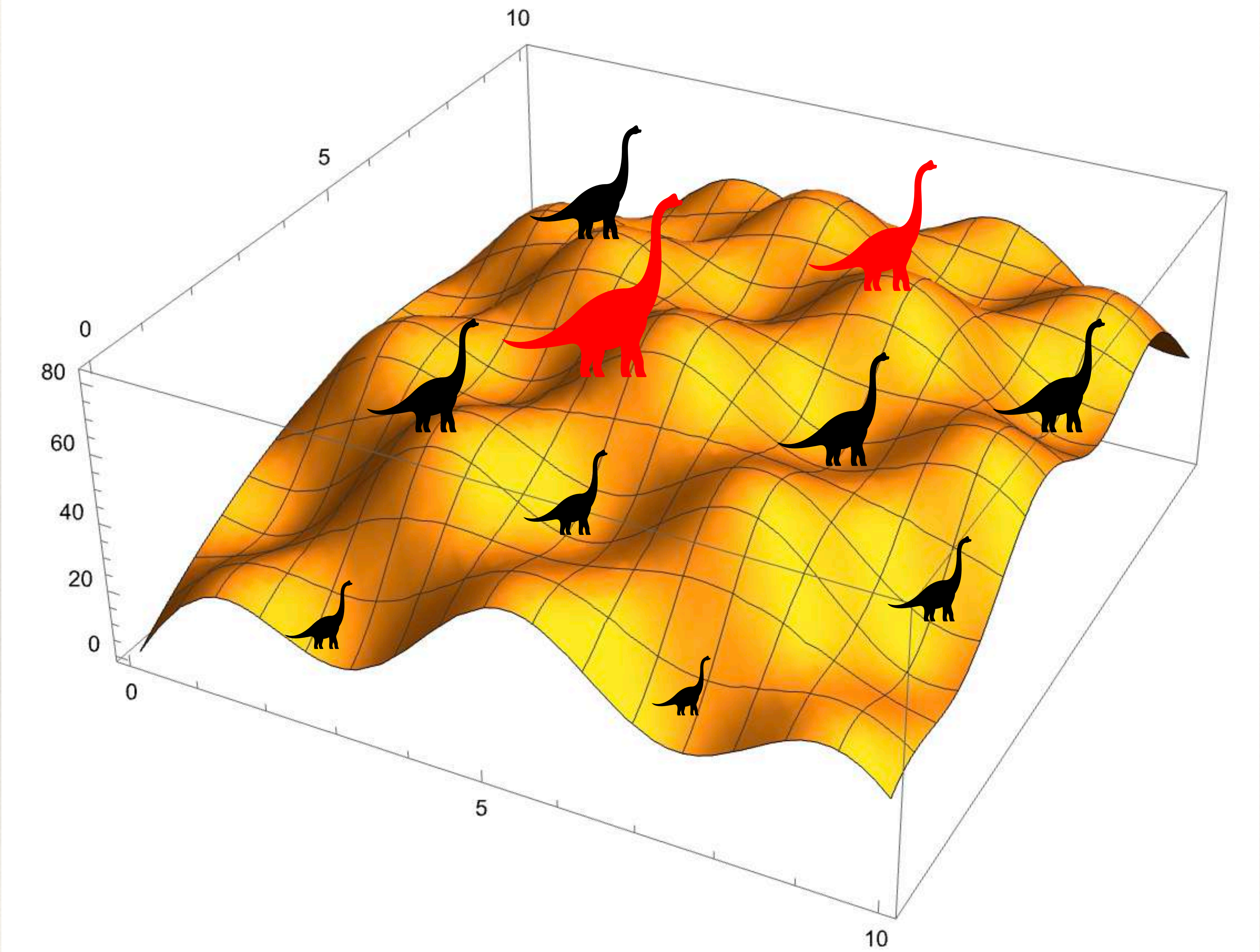
Step 0: population and fitness

Define a population and the **fitness function** ***F***.

Simplest choice: $F = f(x,y)$

Background on Genetic Algorithms

Example: find global maximum to 250 decimal places without using calculus



$$f(x,y) = 12\left(\cos\frac{3y}{2}\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

Step 1: **Selection**

Select pairs for breeding such that the **most fit** individuals can **breed several times**, while unfit ones might not breed at all: e.g. "roulette wheel" based on *ranking k*, with $P_1 = \alpha P_{N_{\rm pop}}$:

$$P_k = \frac{2}{(1+\alpha)}\left(1 + \frac{N_{\rm pop} - k}{N_{\rm pop} - 1}(\alpha - 1)\right)$$

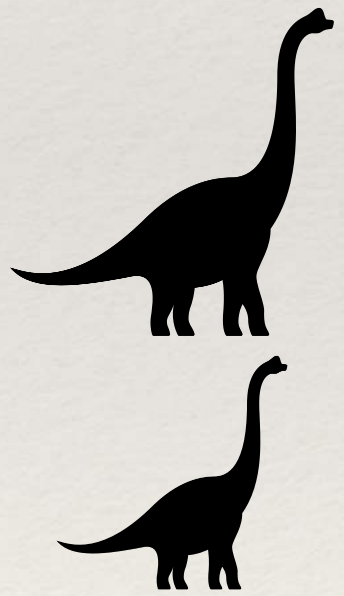
Background on Genetic Algorithms

Example: find global maximum to 250 decimal places without using calculus

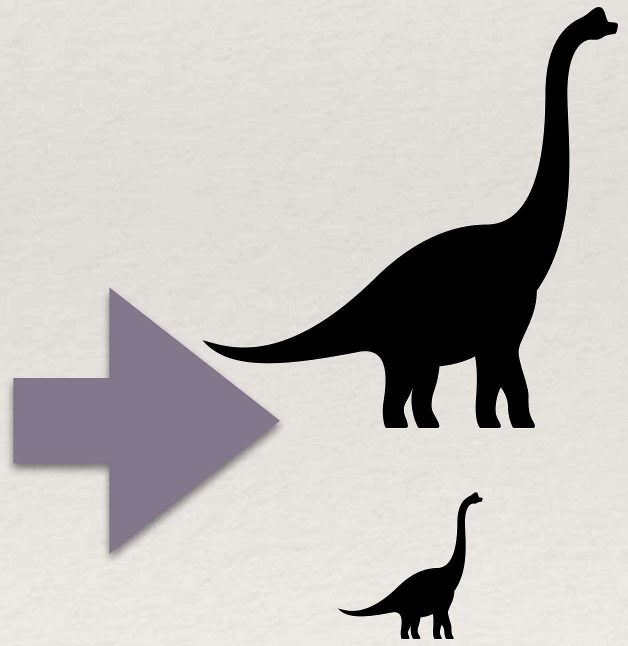
$$f(x,y) = 12\left(\cos\frac{3y}{2}\,\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

Step 2: Breeding

Cut and splice genotypes of breeding pairs somehow (not really crucial how) to make an entirely new population of the same size.



$$g.hij \,|\, kl$$
$$a.bcd \,|\, ef$$



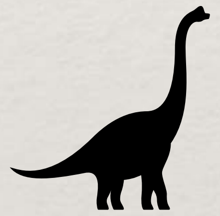
Background on Genetic Algorithms

Example: find global maximum to 250 decimal places without using calculus

$$f(x,y) = 12\left(\cos\frac{3y}{2}\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

Step 3: **Mutation**

Mutation of a randomly chosen small percentage of digits (alleles)



$$a.bcdef'ghi'j\ldots$$

Step 4:

Do the same thing again from step 1.

Background on Genetic Algorithms

Example: find global maximum to 250 decimal places without using calculus

$$f(x, y) = 12\left(\cos\frac{3y}{2}\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$

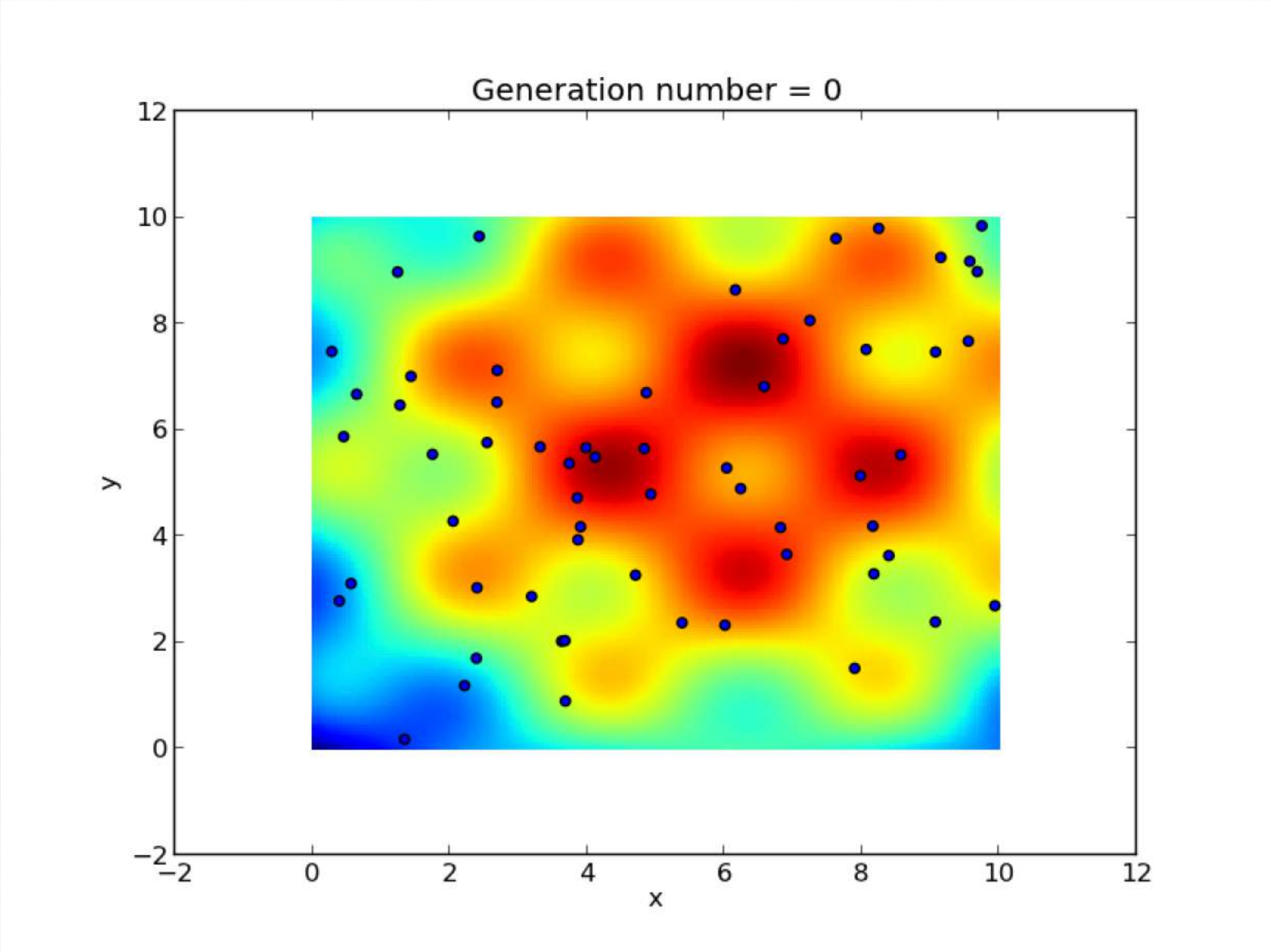
Summary

1. Selection (favours the optimisation);
2. Breeding/crossover (propagates favourable properties);
3. Mutation (prevents stagnation: evolution proceeds by punctuated equilibria)

Background on Genetic Algorithms

Example: find global maximum to 250 decimal places without using calculus

$$f(x, y) = 12\left(\cos\frac{3y}{2}\,\sin\frac{3x}{2} + x + y\right) - x^2 - y^2.$$



Background on Genetic Algorithms

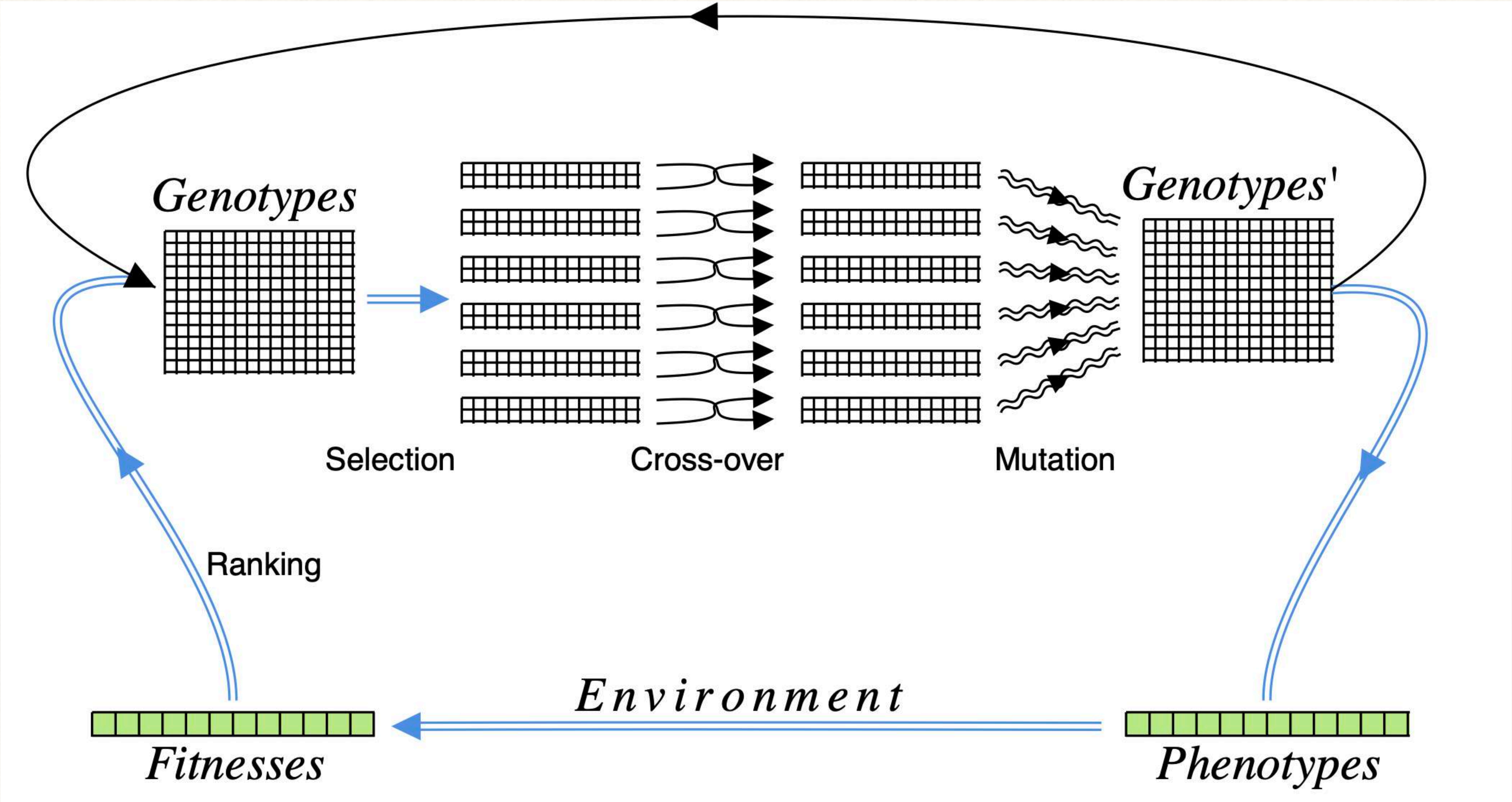
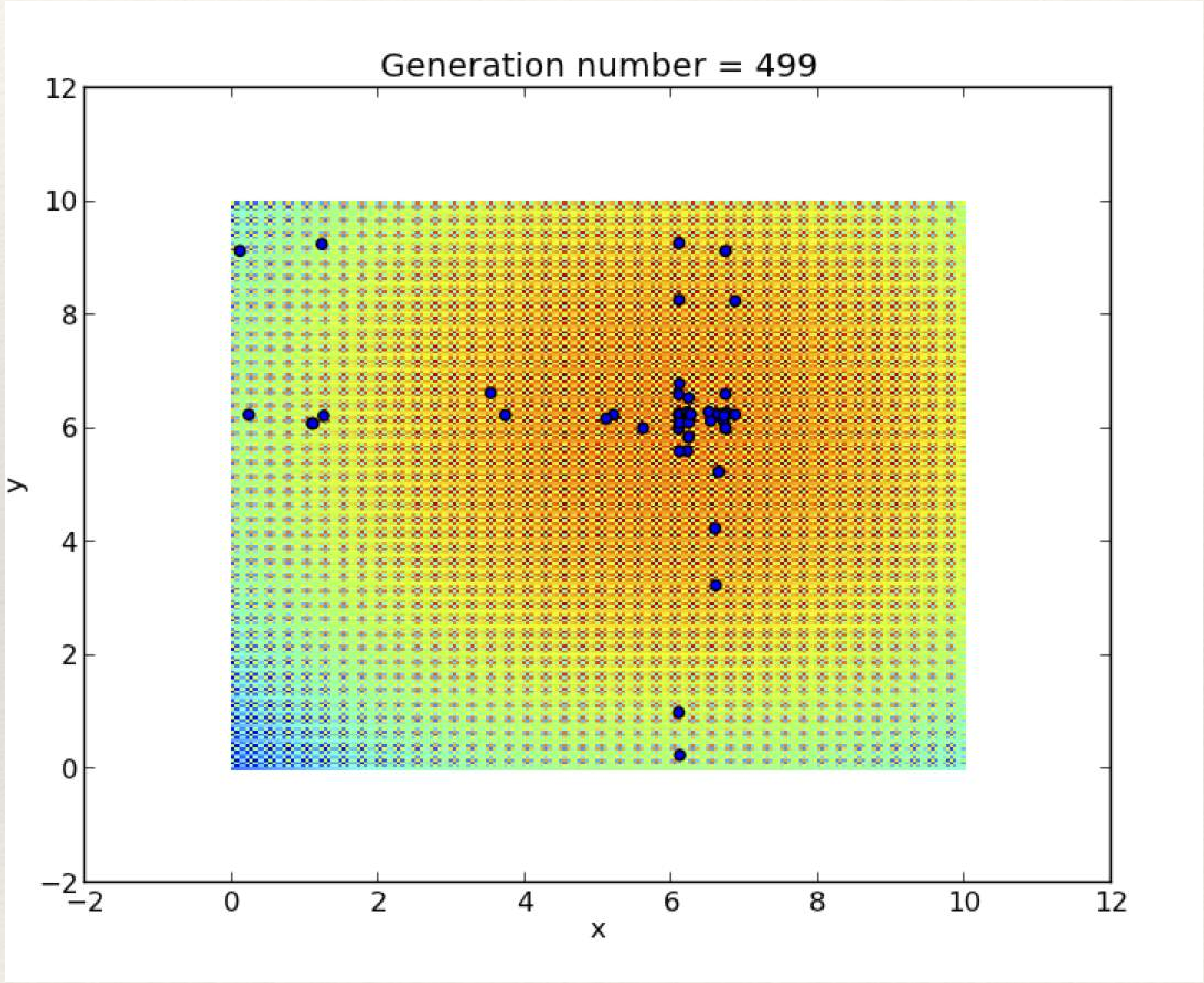


Diagram representing classical GA.

Why do they work?

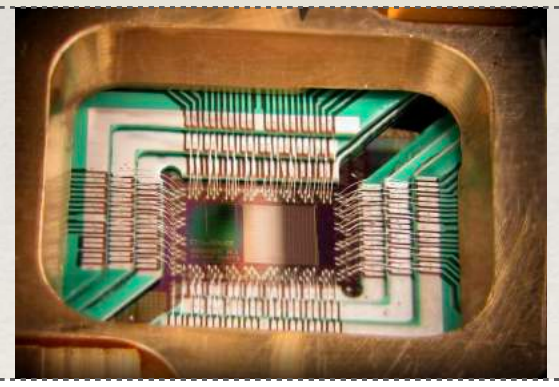
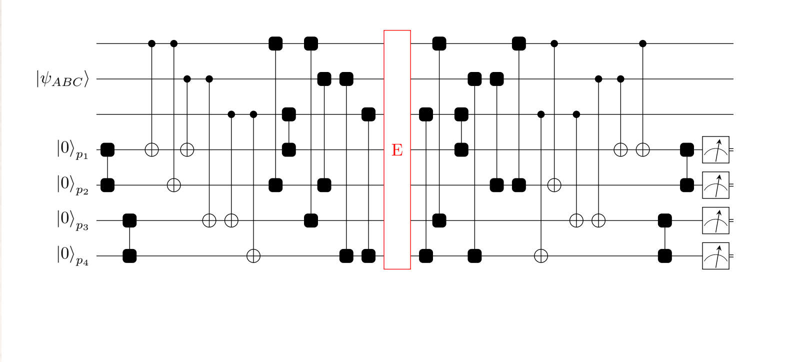
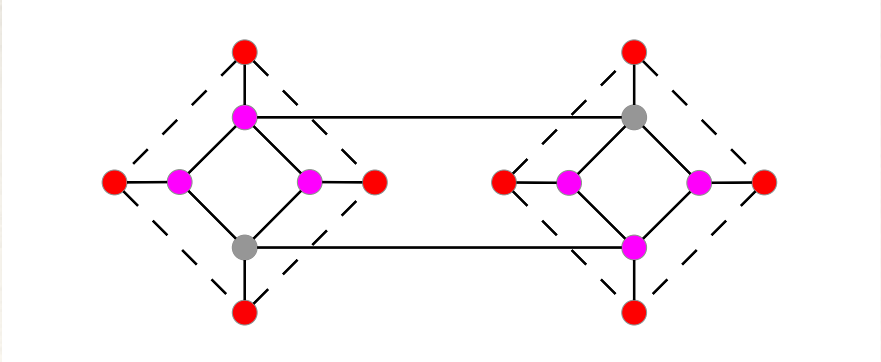
- Holland proposed a probabilistic explanation for the efficiency of genetic algorithms: based on growth rate of "good" schema S , e.g. here $S = 61 * * * 62 * **$
- Holland argues that initial growth of a good schema in the population is exponential
- Selection pushes towards convergence
- Mutation pushes system away from convergence
- Some controversy in 1990s, rehabilitated somewhat by Poli. (Not many good general competing theories)
- Fitness/distance correlation seems to be important
  Holland; David; Jones+Forrest; Collard, Gaspar, Clergue, Escazu



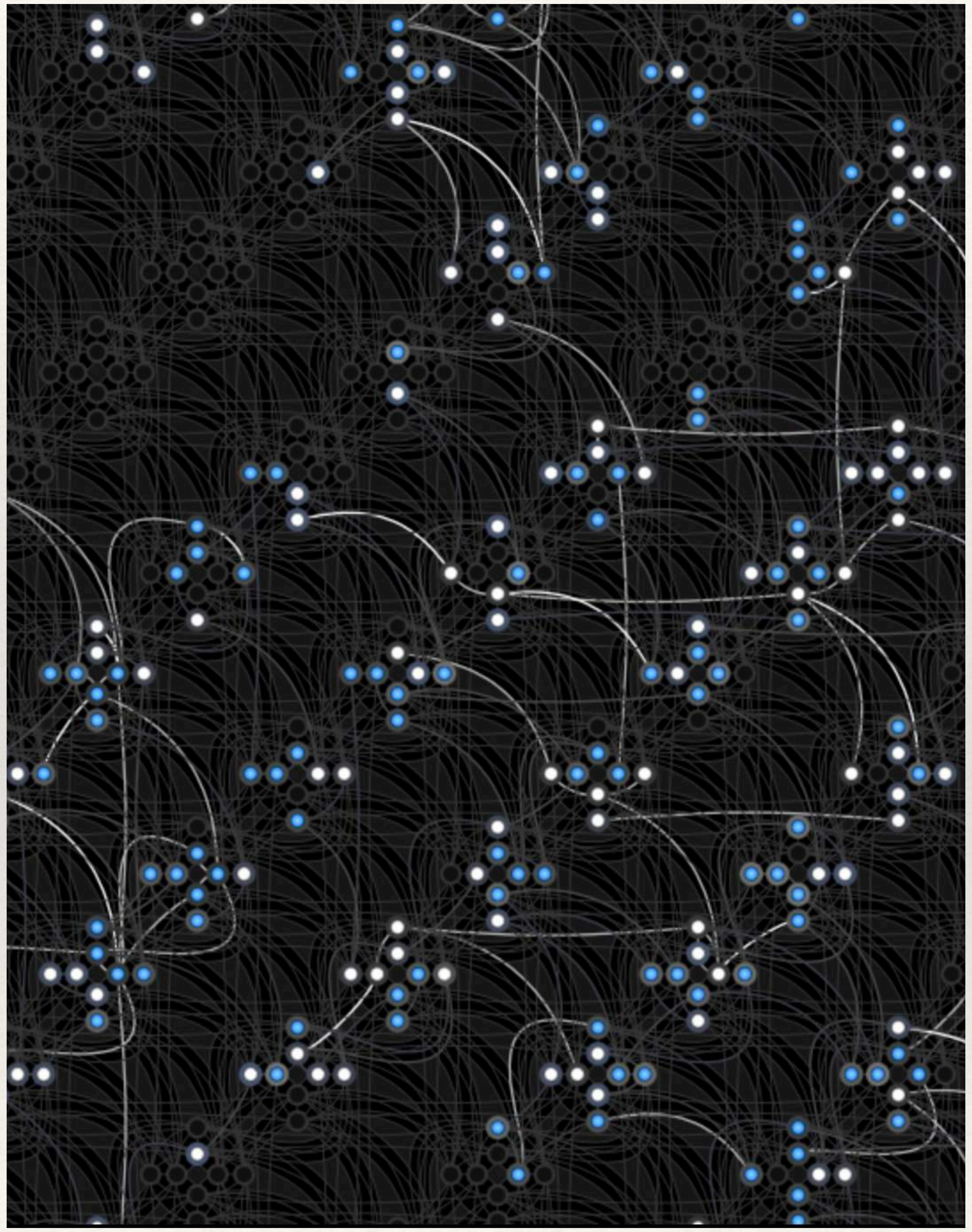
In this example the leading digits of x and y are schemata and get propagated throughout the population

Introduction to Quantum Annealing

Quantum computing has a long and distinguished history but is only now becoming practicable. (Feynman ’81, Zalka '96, Jordan, Lee, Preskill … see Preskill 1811.10085 for review). Two main types of Quantum Computer:

Type	Discrete Gate	Quantum Annealer
Property	Universal (any quantum algorithm can be expressed)	Not universal — certain quantum systems
How?	IBM - Qiskit ~50 Qubits	DWave - LEAP ~7000 Qubits
What?		
		

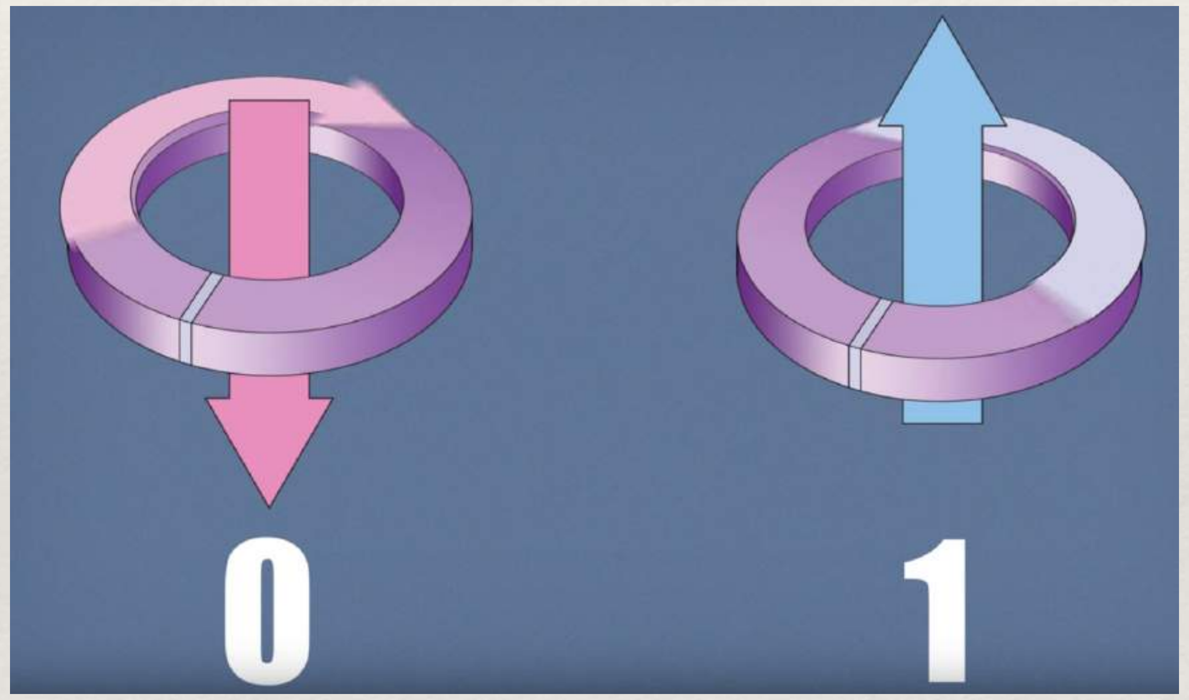
Introduction to Quantum Annealing



Dwave's Advantage_system4.1, Pegasus structure

- **What is?**

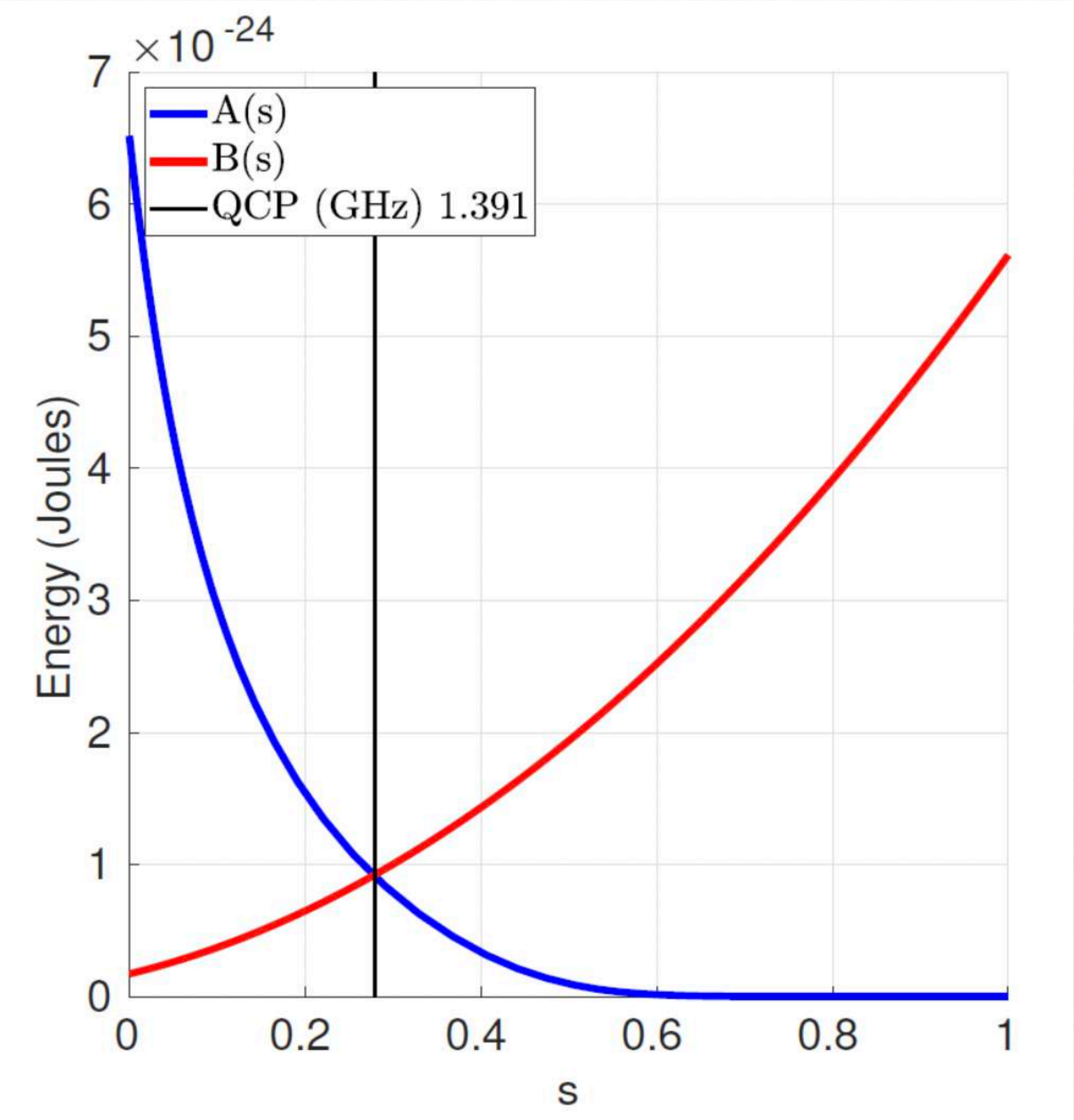
→ **Quantum annealing** (QA) is an **optimisation process** for finding the **global minimum** of a given **objective function** over a given set of candidate solutions (candidate states), by a process using **quantum fluctuations**



- **What kind of problems can we solve?**

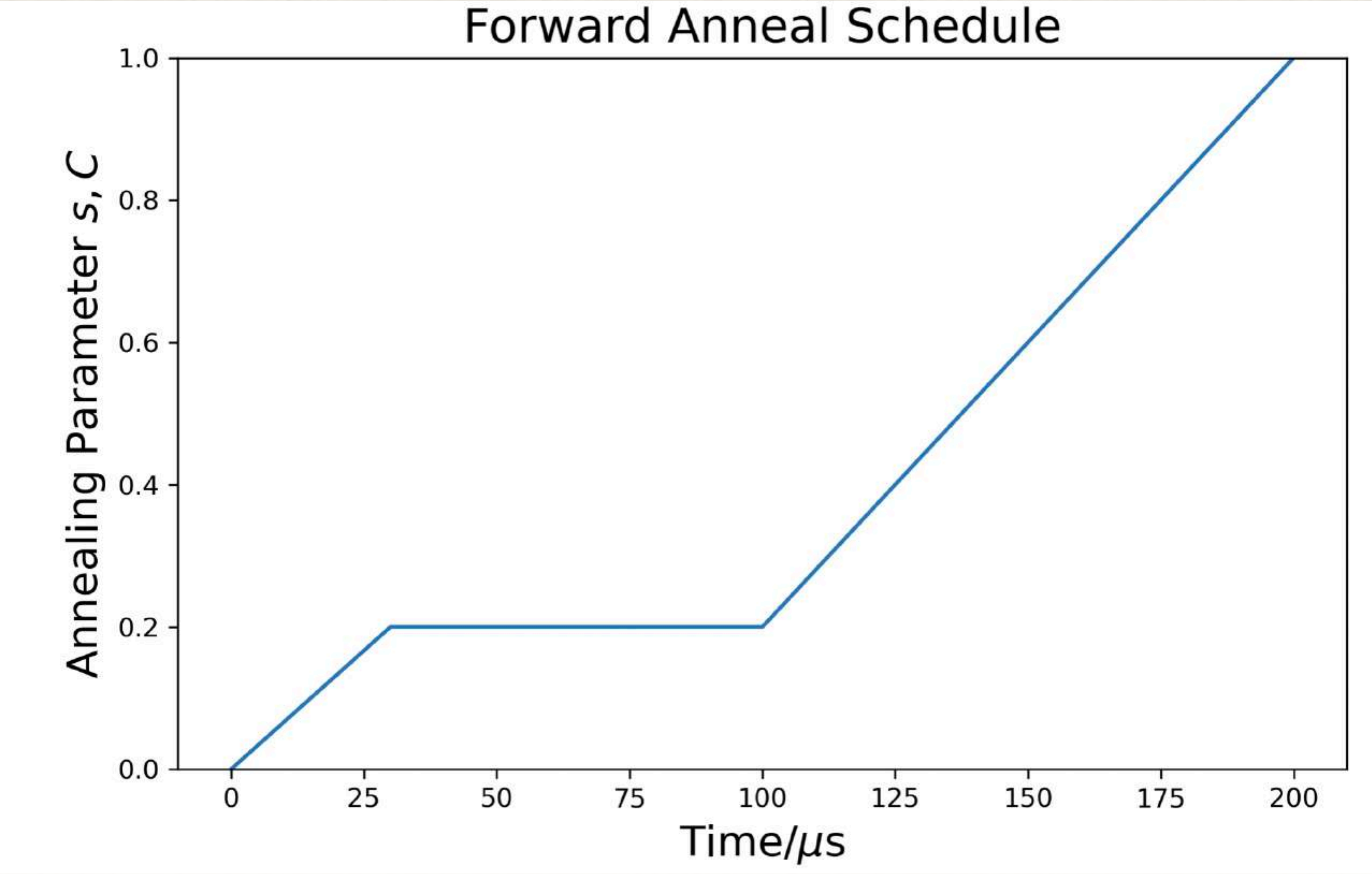
→ Every problem which can be formulated as an **optimisation task** and can be encoded as an **Ising model**.

Introduction to Quantum Annealing



- **How does it work?**

$$H = B(s)\left(\sum_{ij} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z\right) + A(s)\sum_i \sigma_i^x$$



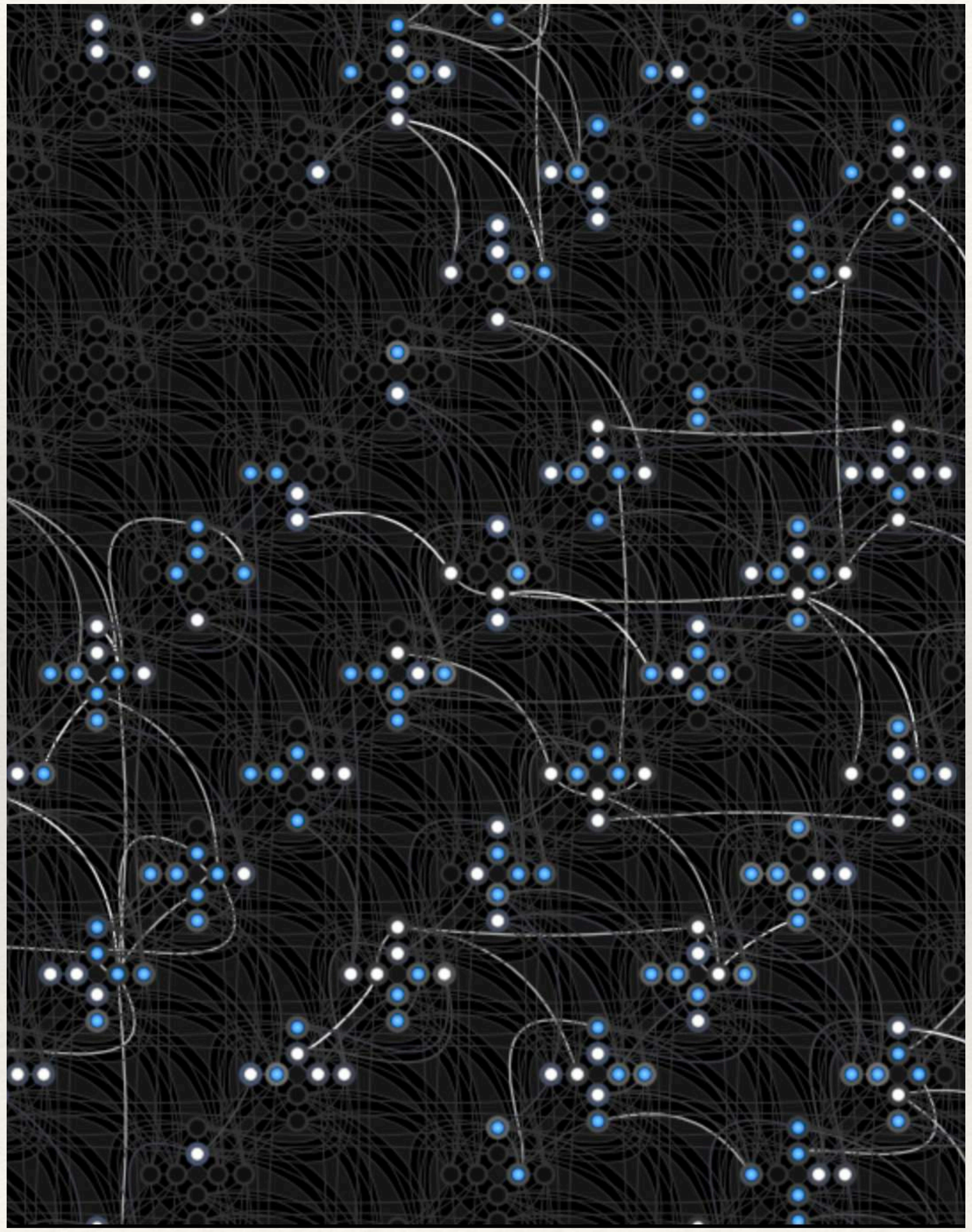
Problem Hamiltonian

Ground state: answer to the problem we are trying to solve

Tunneling Hamiltonian

Ground state: all qubits in a superposition of states

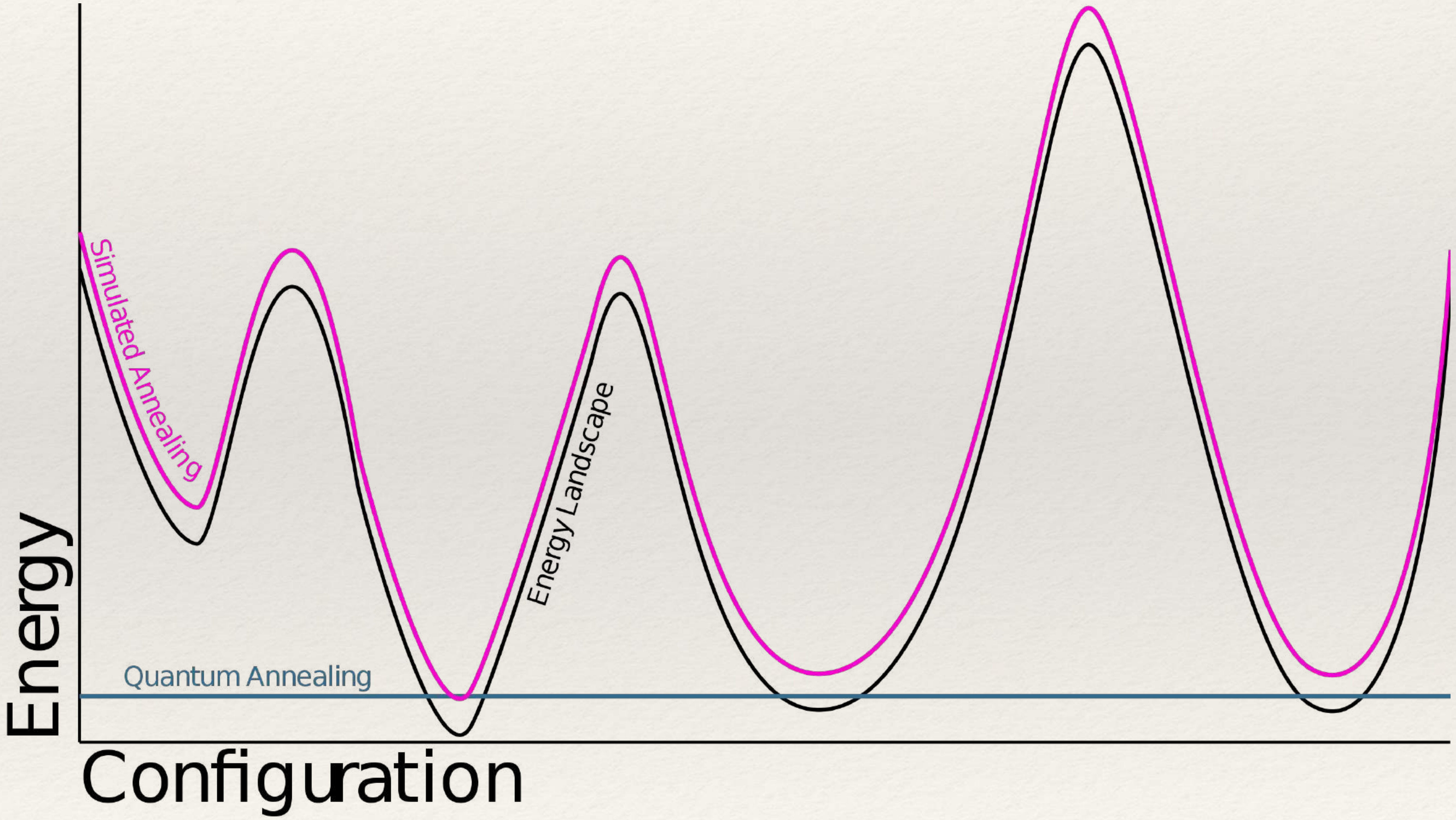
Introduction to Quantum Annealing



Dwave's Advantage_system4.1, Pegasus structure

- **Where is the advantage against classical techniques?**

  → It can find the global minimum by **tunnelling**.



Introduction to Quantum Annealing

- To do this we would simply fill h and J and call the quantum annealer from python as follows:

```
response = sampler.sample_ising(h,J,seed=1234+i,num_reads=3000000, num_sweeps=1)
```

- “response” is a list of [+1,-1,+1,+1 …..] spins ordered by energy

- However the architecture (connectivity of J,h) is limited.

How quantum annealing can improve genetic algorithms?

Genetic Quantum Annealing Algorithm (GQA)

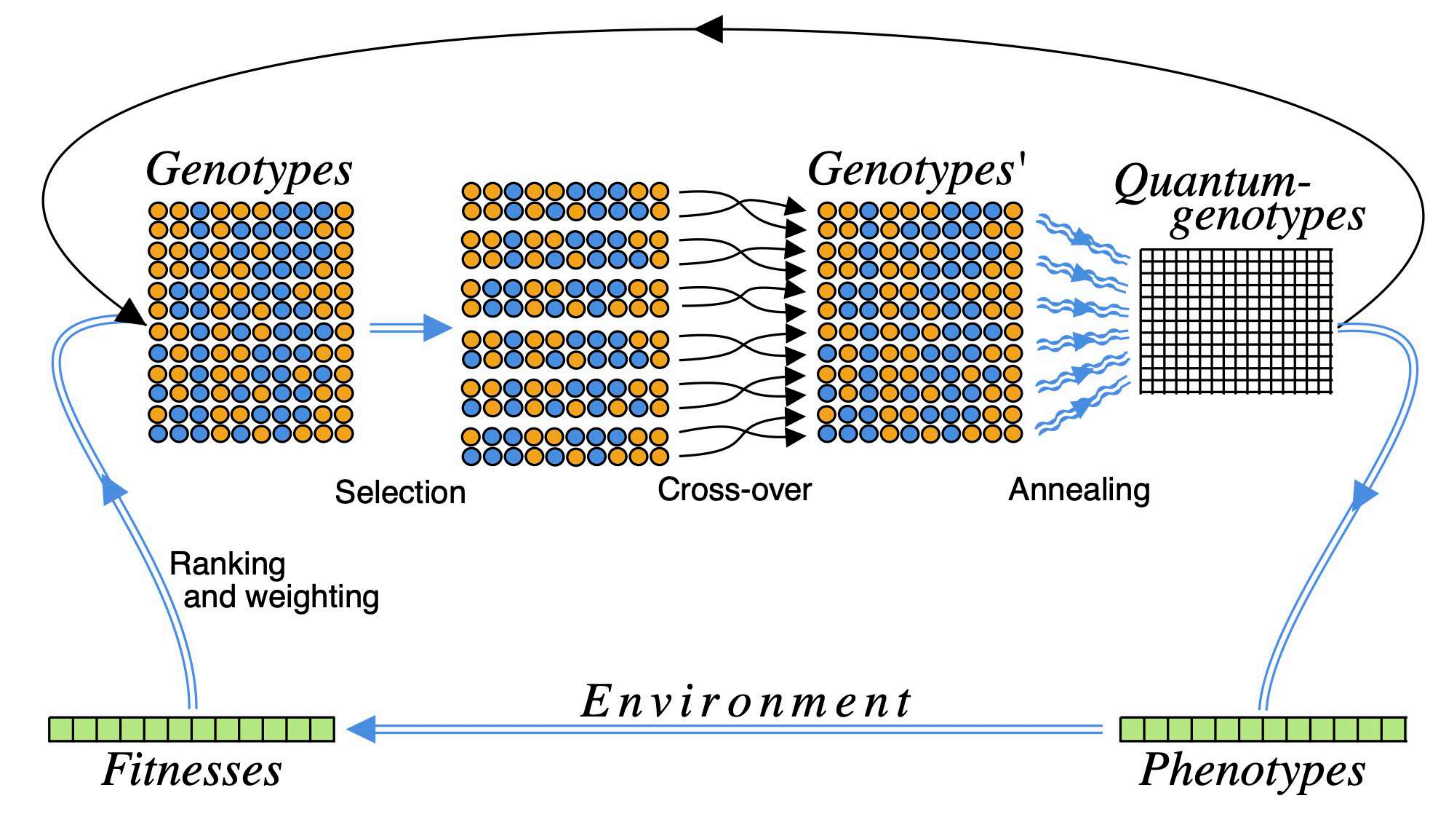
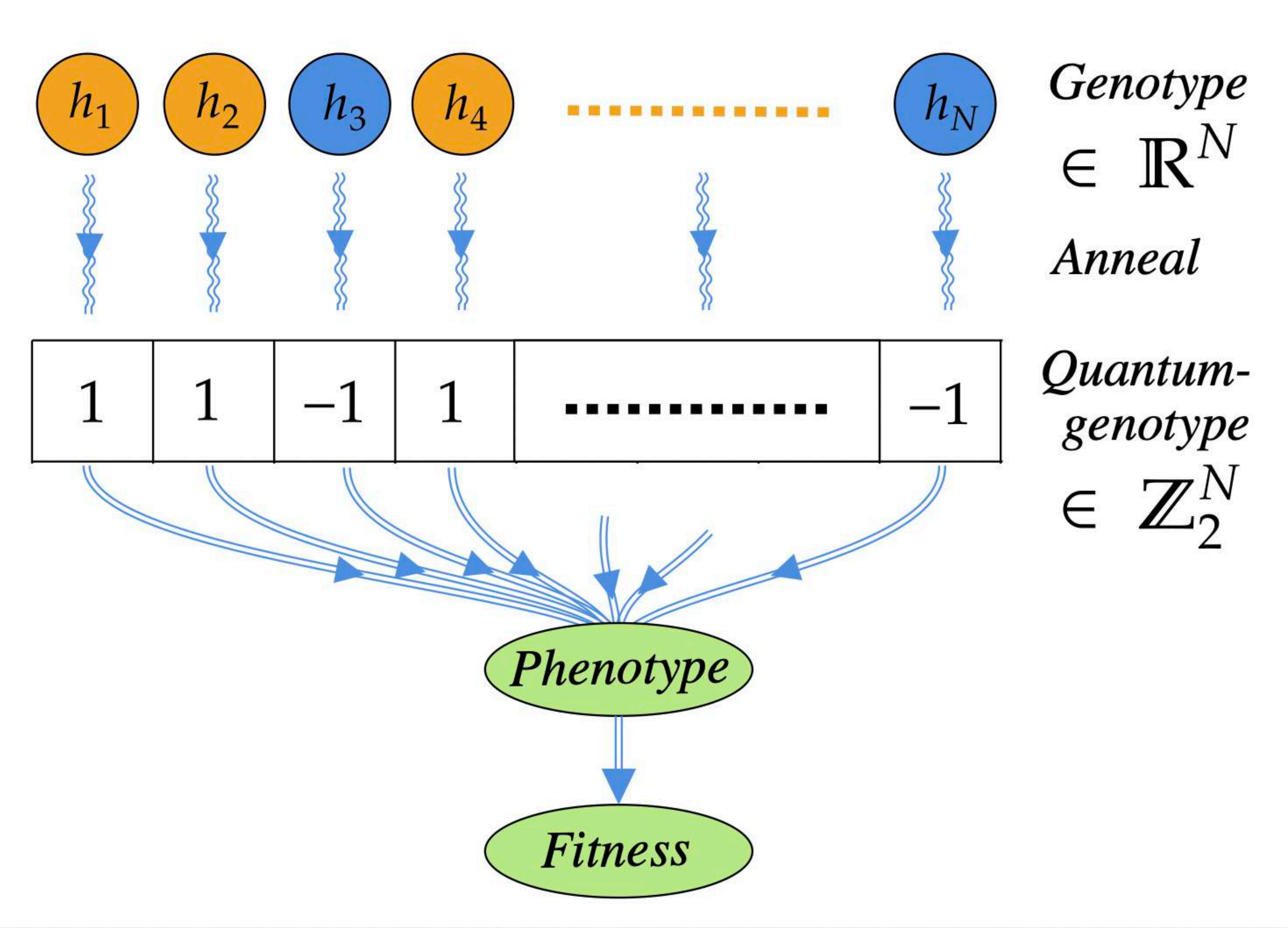


Diagram representing GQAA.

[Abel, LAN, Spannowsky, 2022]

How quantum annealing can improve genetic algorithms?

Classical vs Quantum Genotype



Representation of an individual member of the population in the GQAA.

Fitter individuals have **larger modulus** enforcing their **biasing** more **strongly**.

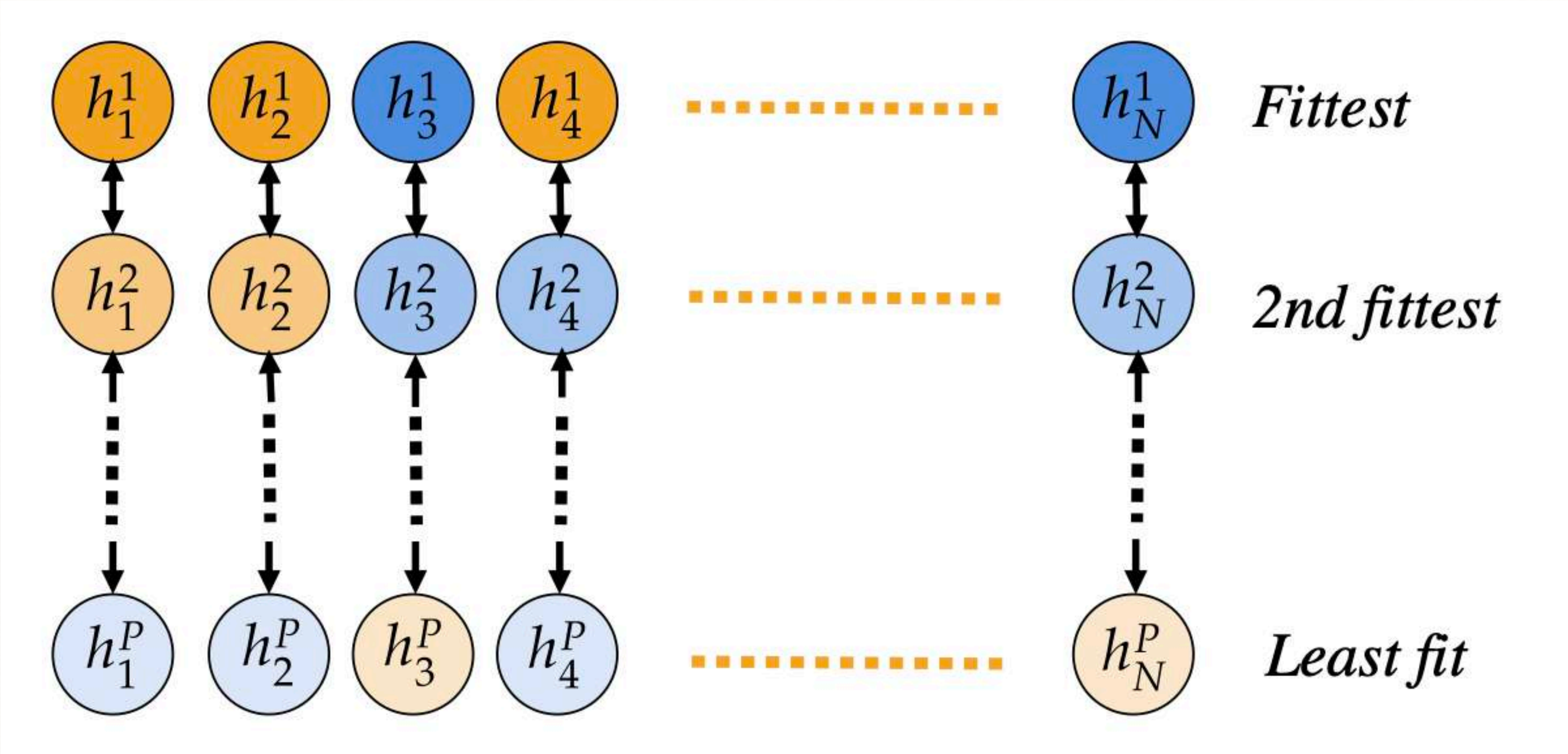
For example the following linear weighting

$$|h_i| = \alpha_p \left(\frac{\alpha - 1}{N_{\text{pop}} - 1} i + 1 \right), \quad i = 0, \ldots, N_{\text{pop}} - 1$$

[Abel, LAN, Spannowsky, 2022]

How quantum annealing can improve genetic algorithms?

Putting the population on the annealer



- The **ranking** is based on the **fitness** of their **parents** (*nepotism*)

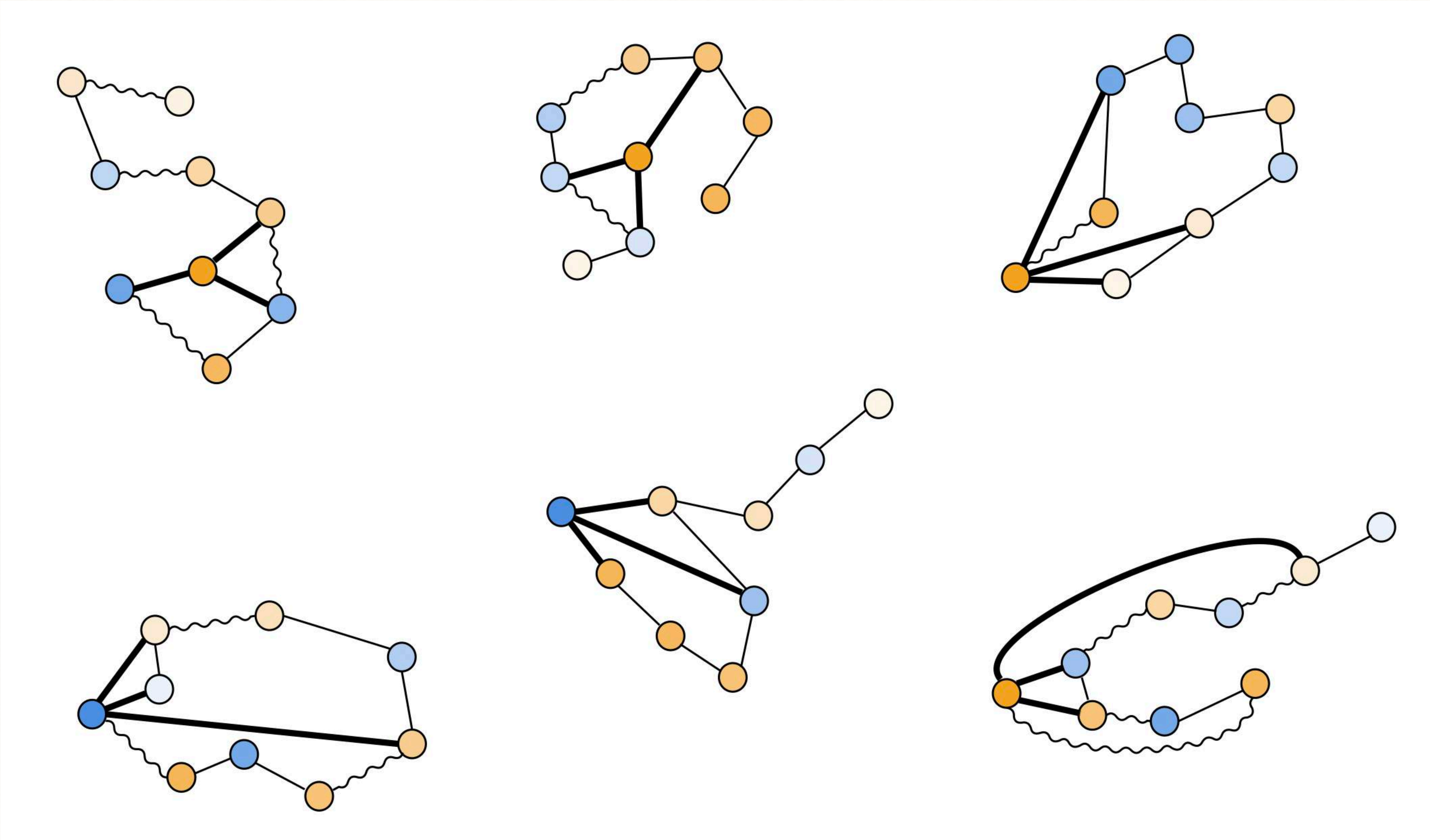
- The quadratic couplings in the quantum annealer allow the individuals to '**see**' the rest of the population

- **Not** an **optimal** configuration: leads to very r**apid convergence** and **stagnation**; the fittest members of the population completely dominate the evolution very early

[Abel, LAN, Spannowsky, 2022]

How quantum annealing can improve genetic algorithms?

Putting the population on the annealer



Example of the topology we used in the annealer.

Wavy lines → Repulsive coupling

Straight lines → Attractive coupling

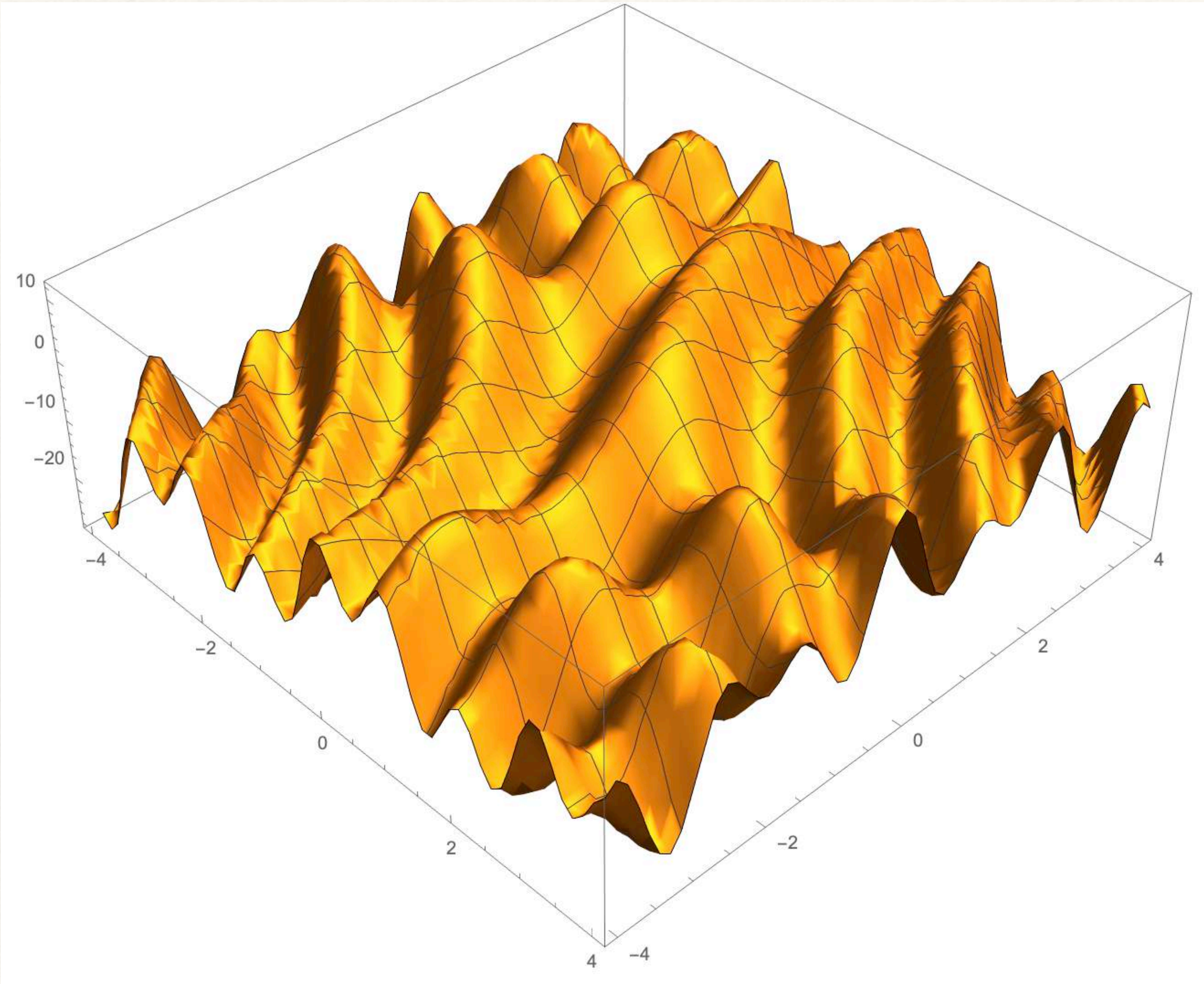
Thicker lines → Attractive coupling between fitter individuals and weaker ones

[Abel, LAN, Spannowsky, 2022]

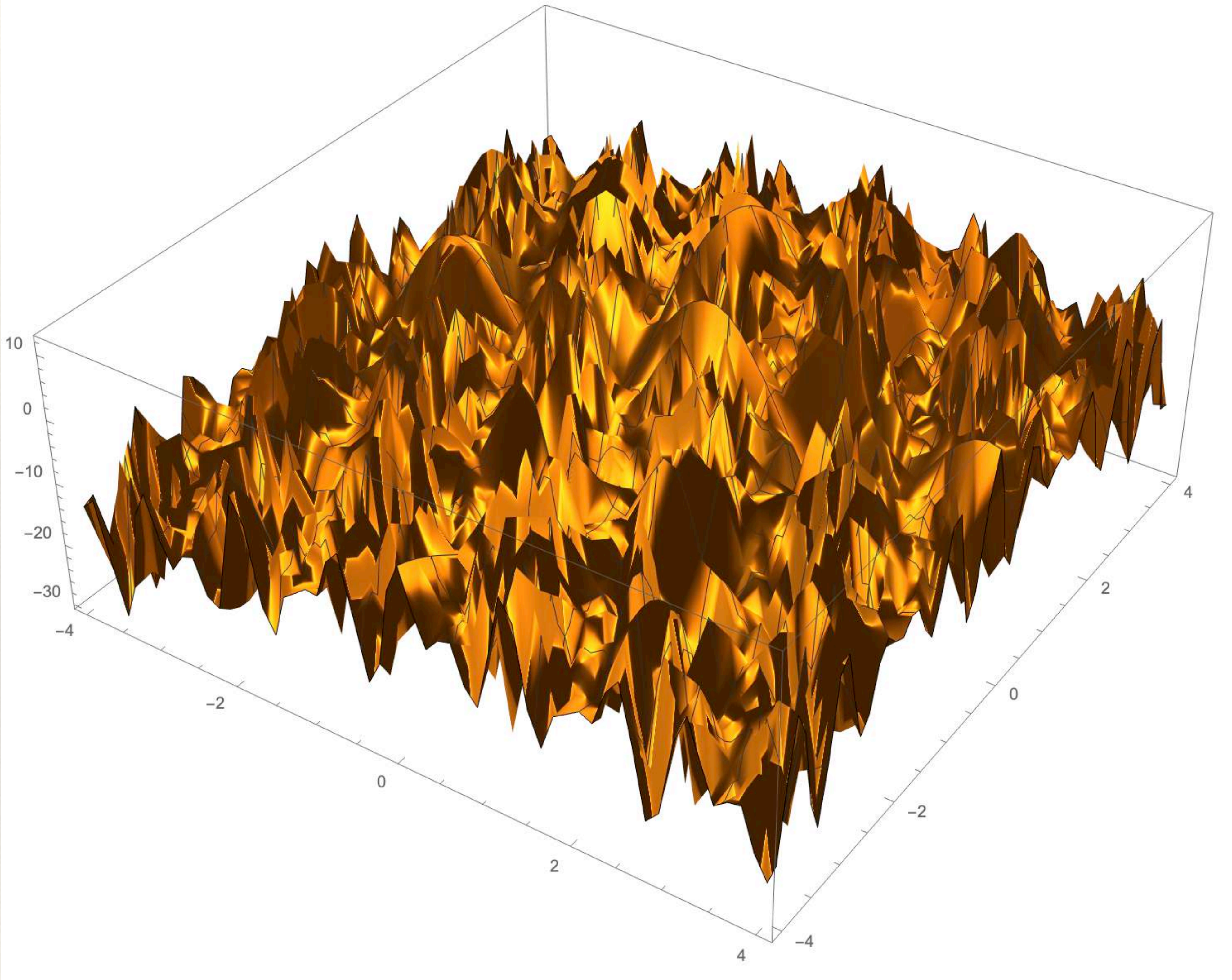
GA vs GQAA

First task

Maximising $U_\kappa(x,y) \;\equiv\; \dfrac{1}{2}\,(x(1-x)\;+\;y(1-y))+\;12\cos(\kappa xy)\sin(2x+y)$ in [-4,4] x [-4,4].



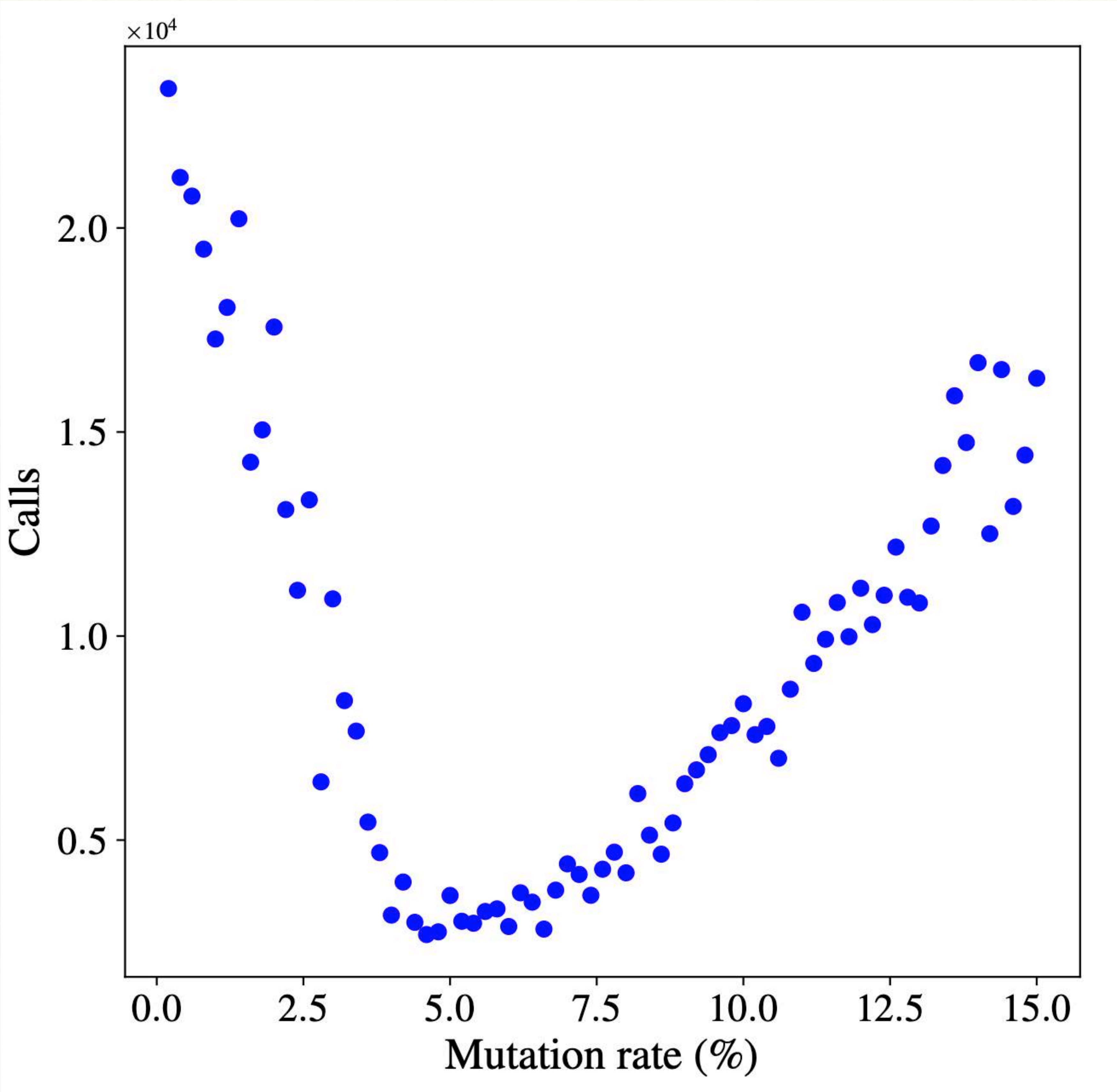
$\kappa = 1$



$\kappa = 20$

GA vs GQAA

Optimising GA

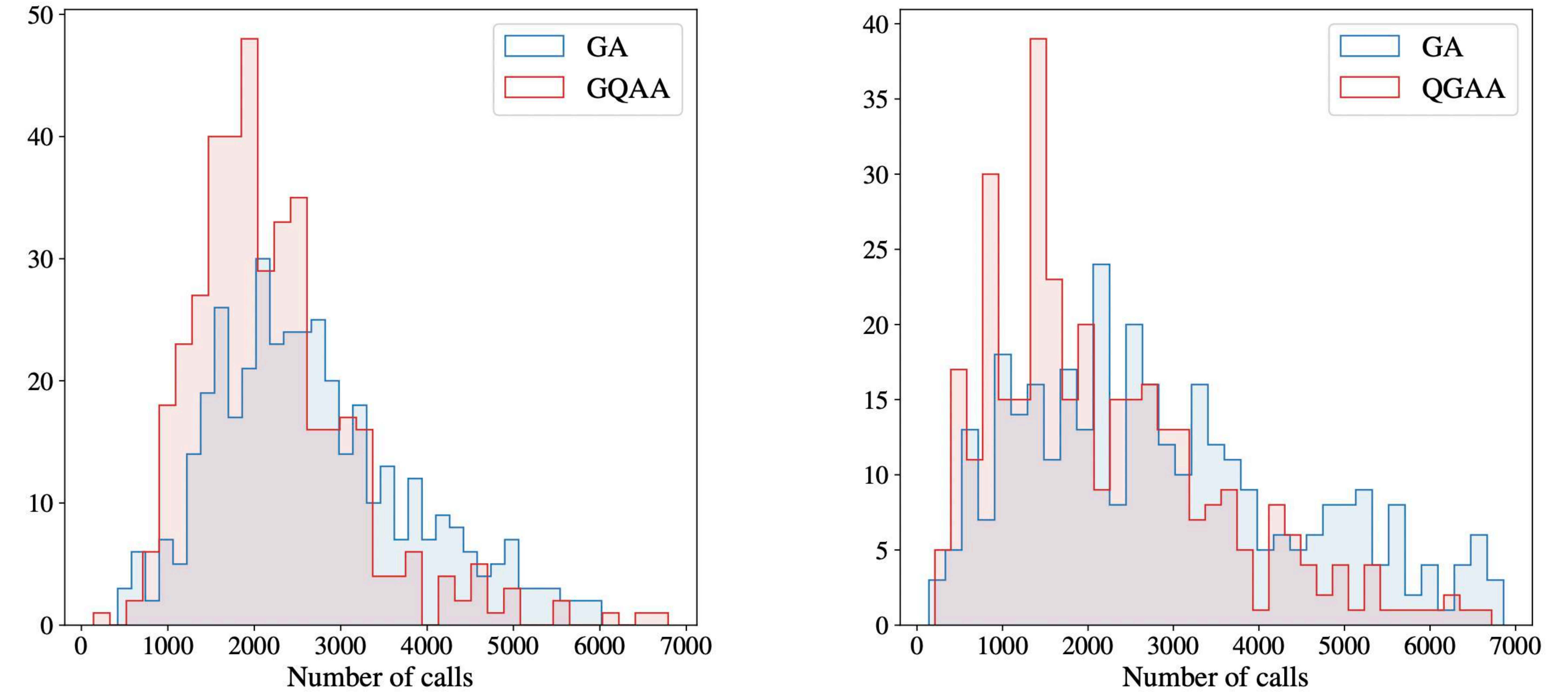


Number of **calls (generation * N_{pop})** required to **find a solution** for different **mutation rates** for the classical GA. For this specific problem the **best mutation rate** is around **5%**.

[Abel, LAN, Spannowsky, 2022]

How quantum annealing can improve genetic algorithms?

Results



(a) $\kappa = 1$ (b) $\kappa = 20$

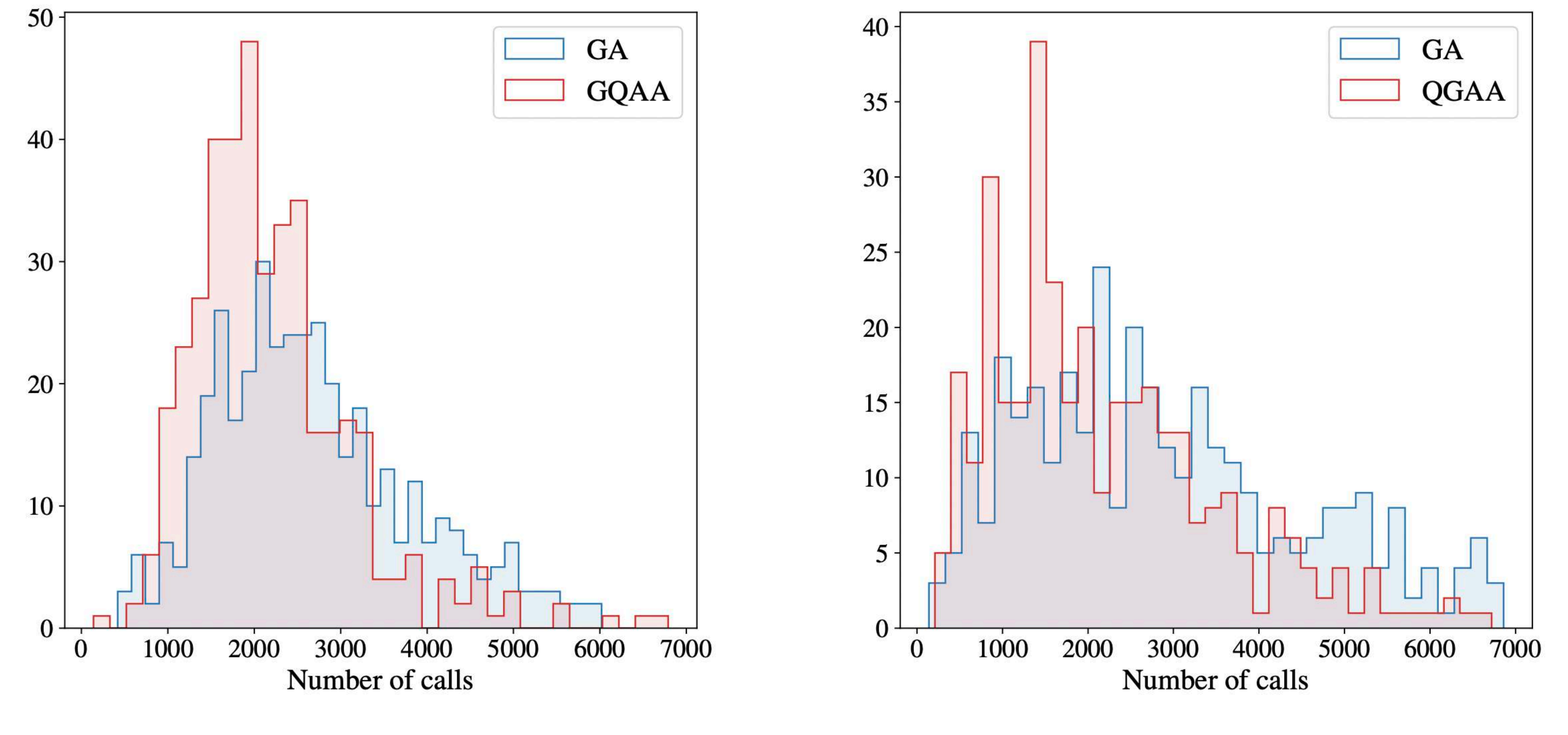
Average number of calls:
GA —> 2690
GQAA —> 2240

Average number of calls:
GA —> 2883
GQAA —> 2186

[Abel, LAN, Spannowsky, 2022]

How quantum annealing can improve genetic algorithms?

Results



(a) $\kappa = 1$ (b) $\kappa = 20$

On average **GA fails** (does not find a solution within the first 7000 calls) in ~**20**% of cases.
This percentage reduces to ~**7**% in the **GQAA** case.

[Abel, LAN, Spannowsky, 2022]

GA vs GQAA

Second task: Taxicab numbers

"Taxicab" numbers are numbers that can be expressed in more than one way as sums of equal powers.

$$\mathrm{Ta}(2) = 1729 = 1^3 + 12^3 = 9^3 + 10^3 \quad \textit{(Ramanujan-Hardy)}$$

$$\mathrm{Ta}(3) = 87539319 = 167^3 + 436^3 = 228^3 + 423^3 = 255^3 + 414^3 \quad \textit{(1957)}$$

$$\vdots$$

$$\begin{aligned}\mathrm{Ta}(6) &= 24153319581254312056344 \\ &= 28906206^3 + 582162^3 \\ &= 28894803^3 + 3064173^3 \\ &= 28657487^3 + 8519281^3 \\ &= 27093208^3 + 16218068^3 \\ &= 26590452^3 + 17492496^3 \\ &= 26224366^3 + 18289922^3 \quad \textit{(2003)}\end{aligned}$$

For Ta(n ≥ 7) only upper bounds are known.

GA vs GQAA

Generalised Taxicab numbers

(k,m,n) numbers are such that $(k, m, n) \equiv x_1^k + \cdots + x_m^k = y_1^k + \cdots + y_n^k$

Unsolved problem in mathematics: (5,2,2) numbers

Does there exist any number that can be expressed as a sum of two positive fifth powers in at least two different ways, *i.e.*, $a^5 + b^5 = c^5 + d^5$?

(3,7,7) and (3,8,8) numbers:

We discovered some apparently **new Taxicab numbers** of this kind with a different technique.
[Abel, LAN, *Fortschritte der Physik*, 2022]

GA vs GQAA

Generalised Taxicab numbers

(k,m,n) numbers are such that $(k, m, n) \equiv x_1^k + \cdots + x_m^k = y_1^k + \cdots + y_n^k$

Unsolved problem in mathematics: **(5,2,2) numbers**

Does there exist any number that can be expressed as a sum of two positive fifth powers in at least two different ways, *i.e.*, $a^5 + b^5 = c^5 + d^5$?

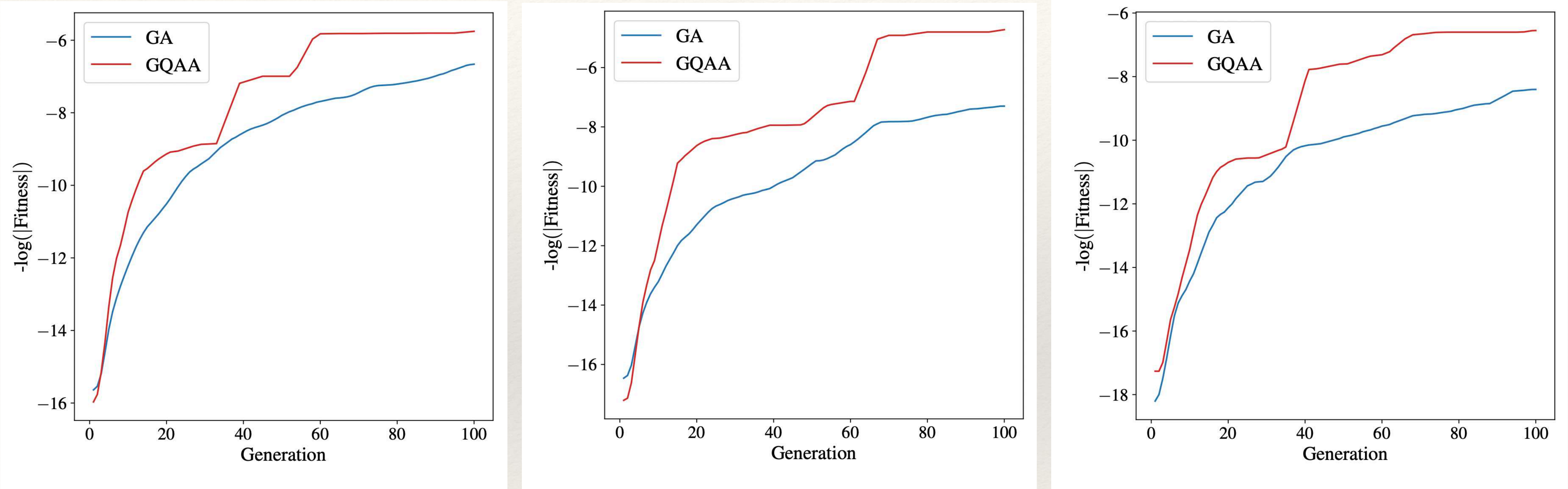
(3,7,7) and (3,8,8) numbers:

We discovered some apparently **new Taxicab numbers** of this kind with a different technique.
[Abel, LAN, *Fortschritte der Physik*, 2022]

We shall focus on the same problem using GQA.

GA vs GQAA

Results on (3,6,6), (3,7,7) and (3,8,8) numbers



(a) Taxicab (3,6,6)   (b) Taxicab (3,7,7)   (c) Taxicab (3,8,8)

~ one order of magnitude difference in the the fitness of the fittest individual after 100 generation

[Abel, LAN, Spannowsky, 2022]

Conclusions and Outlook

- We developed an hybrid technique using genetic algorithms and quantum annealing

- We find the algorithm to be significantly more powerful on several simple problem than a classical GA

- Apply this technique to physical problems (*e.g.* string theory landscape,…)

Thanks for your attention