Hybrid PSO-GA for mixed continuous-discrete problems

Presentation for the exam of *Global and Multiobjective Optimisation*

Matteo Fadelli

Continuous optimization

- set of solutions is **uncountably infinite**
- for many classes of problems *polynomial* algorithms exists
- very effective methods in practice when *other properties* holds (e.g. differentiability)

Discrete optimization

- set of solutions is typically **finite**
- many problems of this kind are provably *NP-hard*
- often hard to solve in practice, heuristics are commonly used

Continuous optimization	**Discrete** optimization

Some applicable metaheuristics

Continuous	Discrete
Particle Swarm Optimization	Genetic Algorithm
Differential Evolution	Evolution Strategies
Explicit methods (e.g. Estimation of Distribution Algorithms)	Simulated Annealing

but many have variations for the other type

Mixed continuous-discrete optimization problems

$$\min_{\vec{u},\vec{v}} f(\vec{u},\vec{v}) \quad \text{where } \vec{u} \in D \text{ and } \vec{v} \in C$$

real valued function

$f : D \times C \to \mathbb{R}$

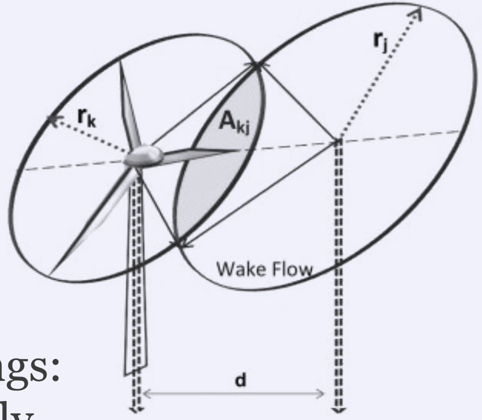
continuous space

usually compact subset of $\mathbb{R}^m$

discrete space

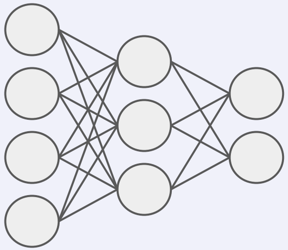
could be a finite subset of $\mathbb{R}^n$ (but in general even categorical)

Mixed problems in the wild - examples



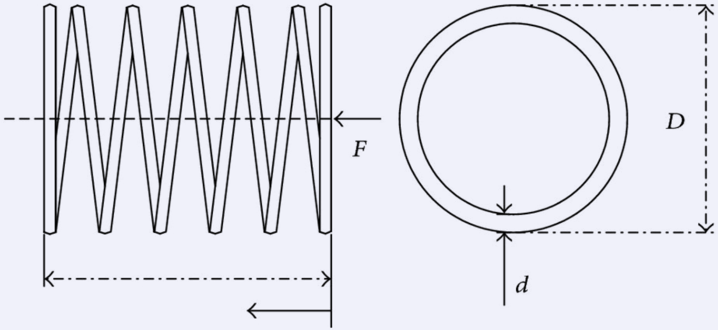
in an engineering or industrial settings:

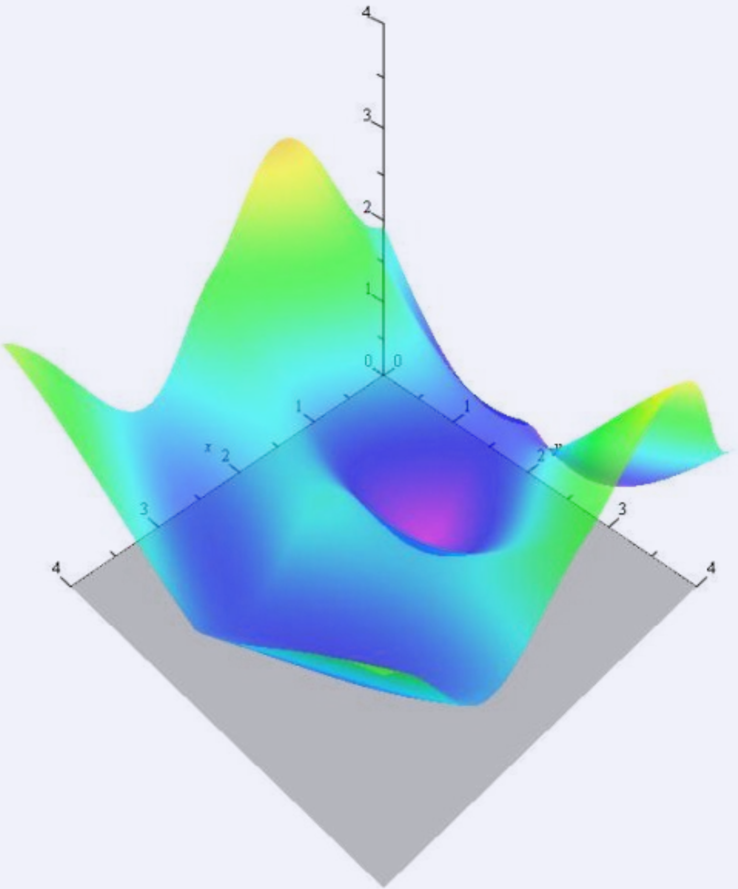
- variables that can be tuned freely (**continuous**)
- counting variables (**naturals**)
- variables that can assume only certain values (**discrete**)

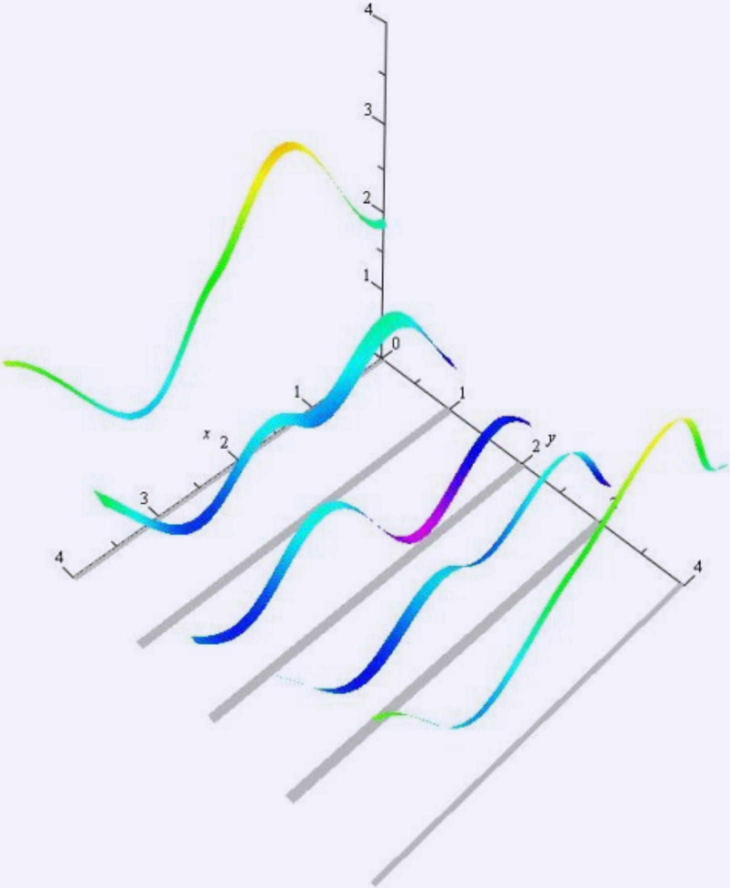


when optimizing NNs:

- learning rate (**continuous**)
- neurons in a layer (**integers**)
- activation function (**categorical**)

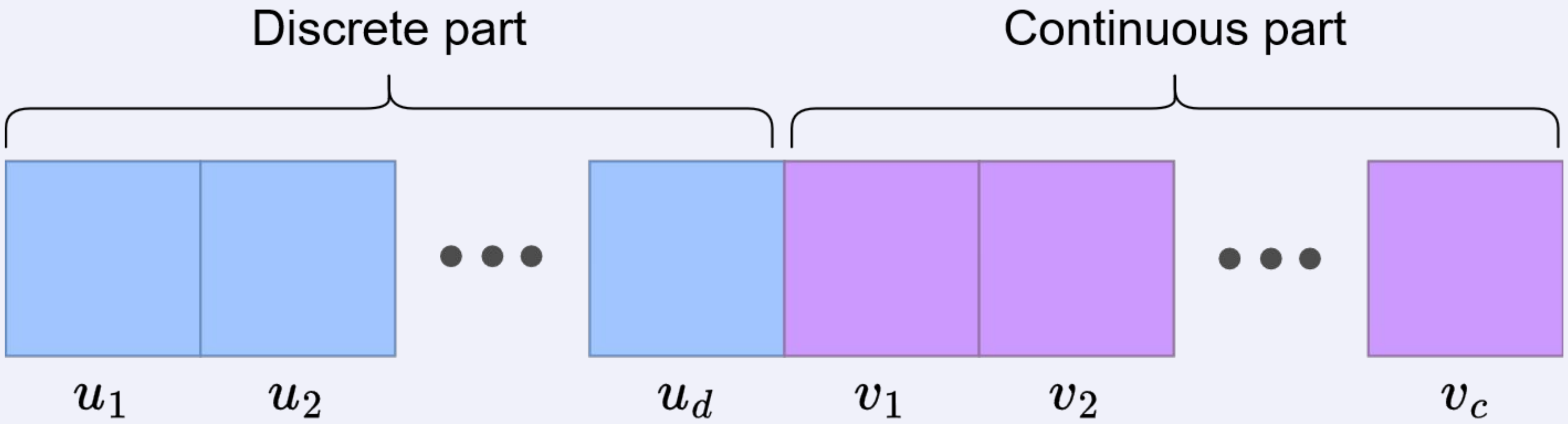




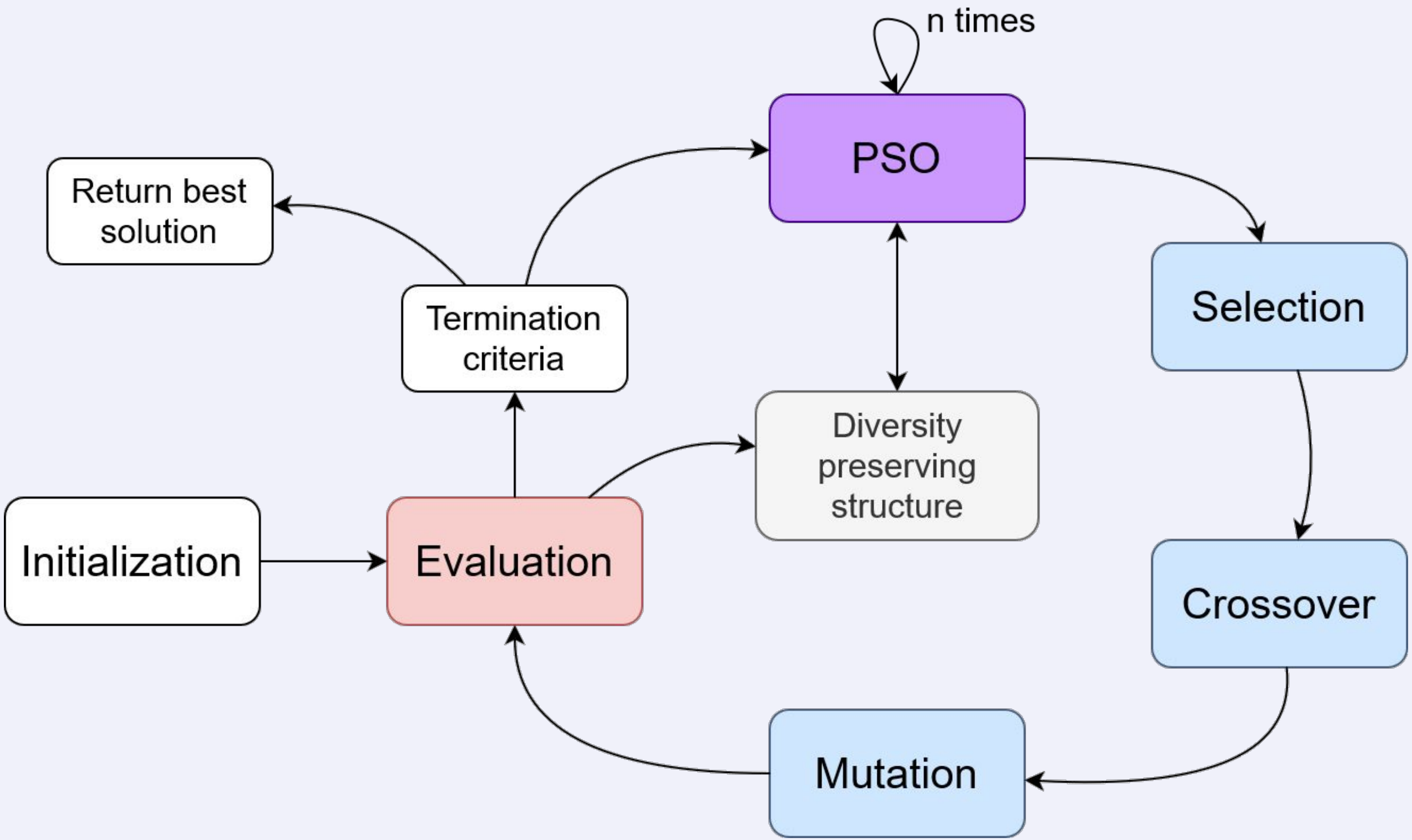


some properties like continuity or differentiability may be preserved

Variable encoding

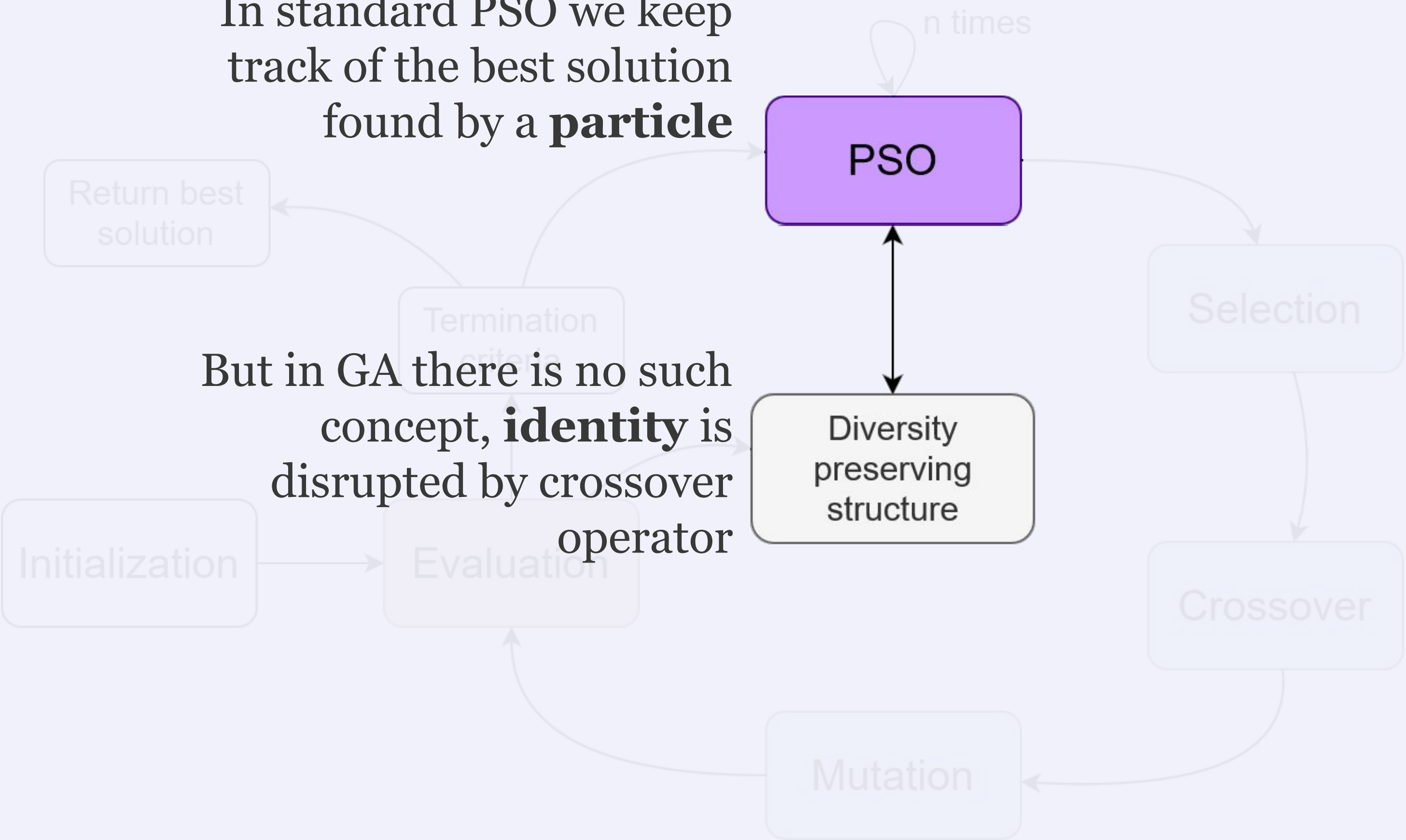


General schema of the proposed algorithm



In standard PSO we keep track of the best solution found by a **particle**

But in GA there is no such concept, **identity** is disrupted by crossover operator



Diversity preserving structures

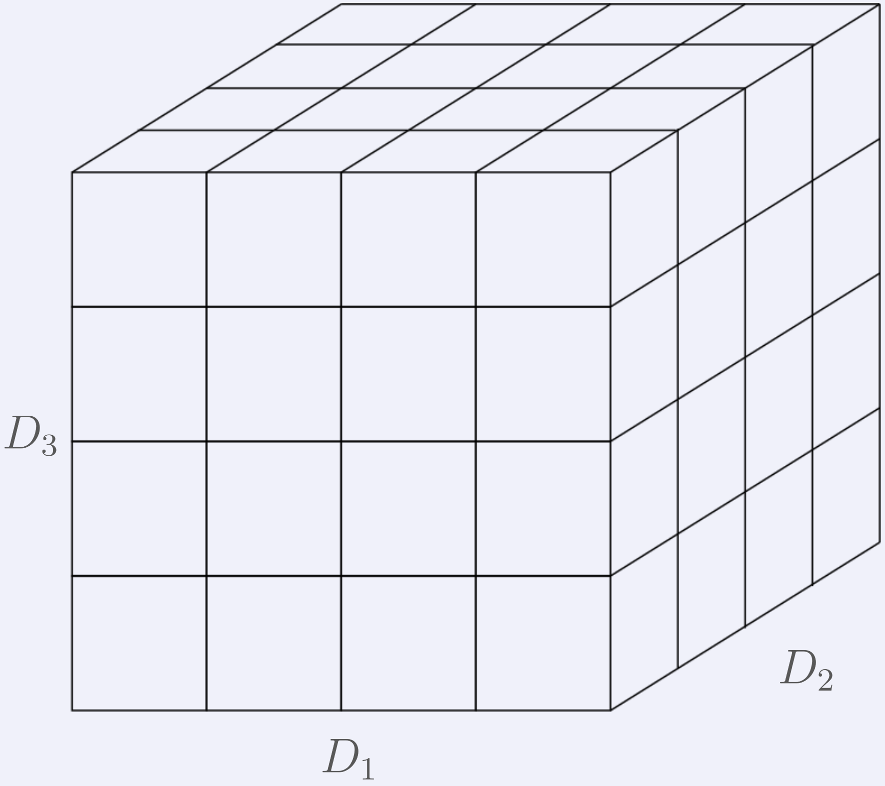
Matrix-like structure

Discrete space is divided into **hypercubes**

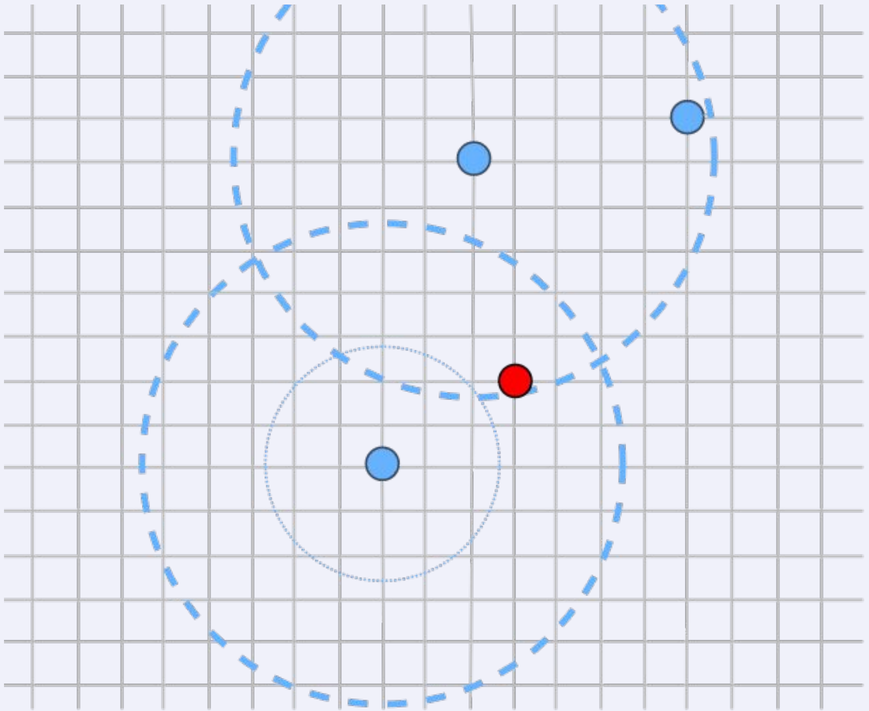
For each hypercube we keep track of the **continuous part** of the best particle with discrete part in the hypercube

During **PSO**, given a particle, the best particle component is the one in the matrix that is in the **same hypercube**

(+) fast, easy to implement
(-) feasible only for low dimension, static



Diversity preserving structures



List-like structure

A list of **fixed length** is maintained

An element **replace** another if the **distance** between their discrete part is small and it has a better value.

An element replace a bad element in the list if it is **good** enough and **sufficiently far** away from the elements in the list

During **PSO**, given a particle, the best particle component is the one in the list that is **closer** in their discrete part

(+) dynamic, tunable
(-) many parameters to tune, costly

Genetic Algorithm adjustments

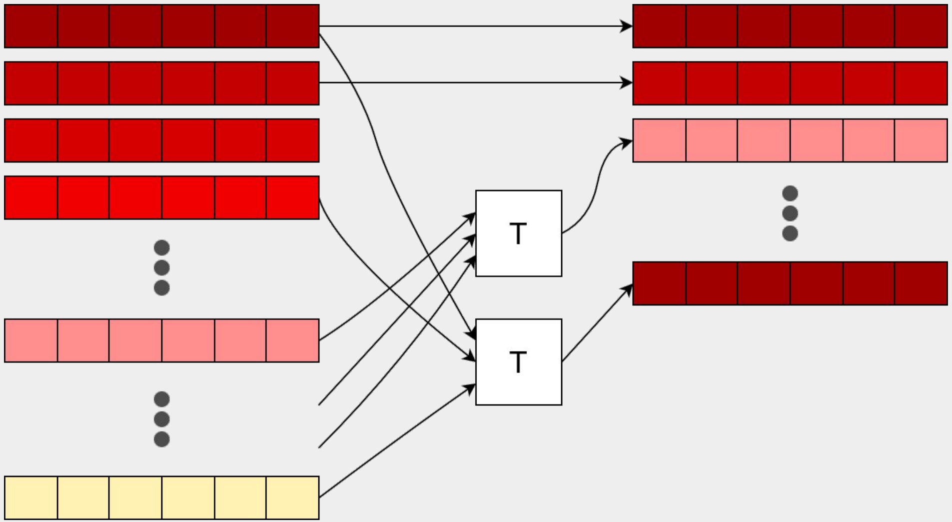
Selection → no adjustments needed w.r.t. standard GA

Crossover → standard GA can be used on the *whole* encoded vector but some improvement can be done

Mutation → two possible options: only for the discrete part or for both continuous and discrete part

Selection - implemented

- the best **1/5th** of the population is **directly** selected
- **tournaments** of **size 3** among the whole population decides the other individuals to be selected
- the number of individuals to be selected is equal to **half** the population

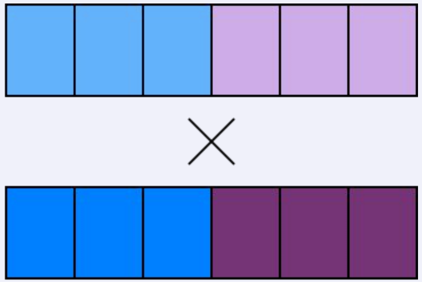


Crossover - implemented

- the best **1/10th** of the population is **preserved**
- then two parents are selected uniformly from the **whole** selected **population**. A crossover is performed, randomly selected among the three types available, generating two children
- this is iterated until we reach a population equal to the **initial population minus 1/10th** of the initial population

There are **three types** of crossovers:

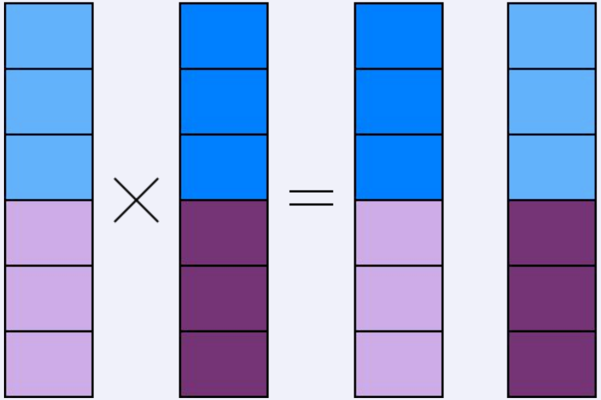
❏ *discrete-continuous* crossover
❏ *double one point* crossover
❏ *uniform* crossover



Discrete-continuous **crossover**

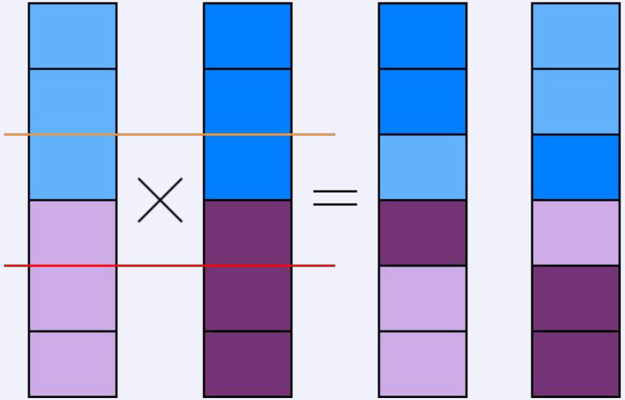
One children inherits the **discrete part** from the *first parent* and the **continuous part** from the *second parent*.

The role of the parents is switched for the other children.



Double one point **crossover**

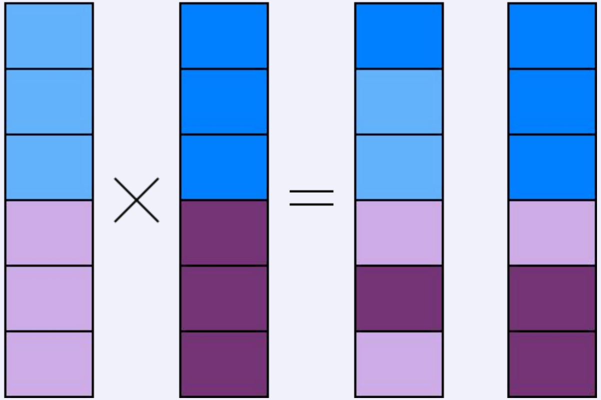
One point is selected for the **discrete part** and another for the **continuous part**. *One point crossover* is performed for both parts.



Uniform **crossover**

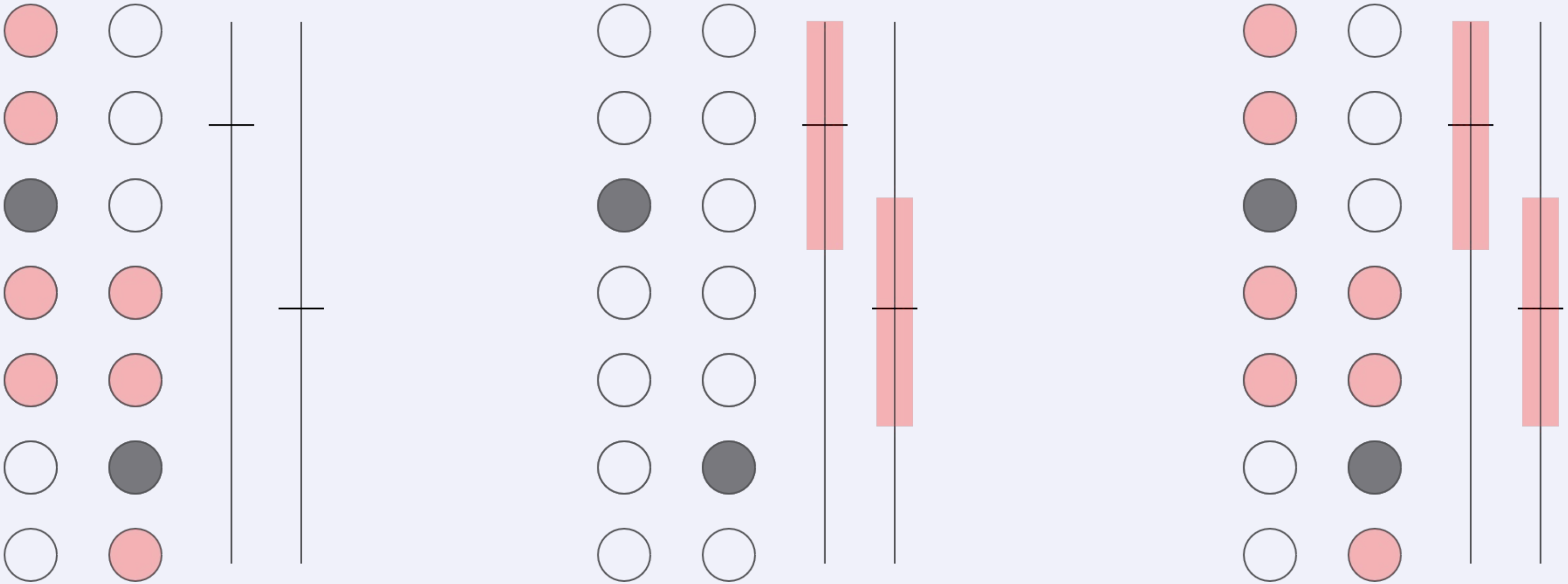
A *classic uniform crossover* is performed on the first parent (the whole encoded vector) with probability ⅓ of selecting the gene from the second parent.

The same is done for the second parent with the roles reversed.



Mutation - implemented

- **1/10th** of the population is preserved **without mutation** (the one that avoided crossover)
- the rest of the population is mutated by one of the following **mutations**:
  - **discrete** only mutation: each discrete variable is mutated, with a certain probability, of a small step (if total ordering) or randomly (if categorical)
  - **continuous** only mutation: each continuous variable is mutated, with a certain probability, of a small epsilon
  - **both** of them together (possibly with smaller step/epsilon)
- the preserved population is **copied** and then mutated using the **discrete** only mutation.

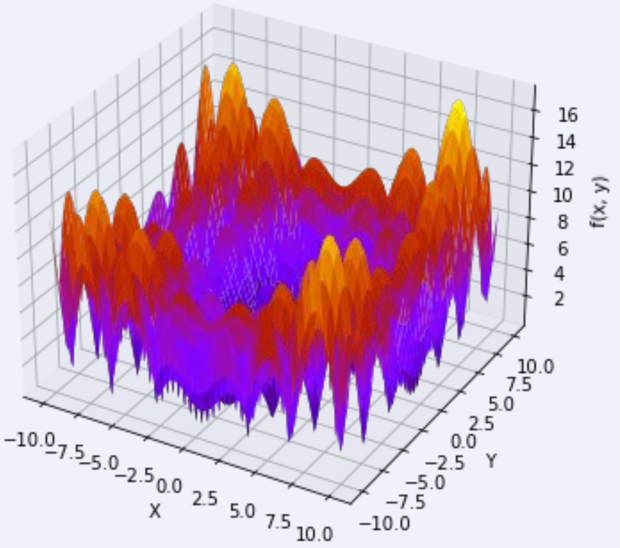


Discrete-only ϵ mutation Continuous-only ϵ mutation Continuous and discrete

Experimental results

Artificial benchmark functions

- Continuous benchmark functions with **half** of the **variables discretized**
- **Hard to optimize** even in the continuous case, minimum in $\vec{0}$.

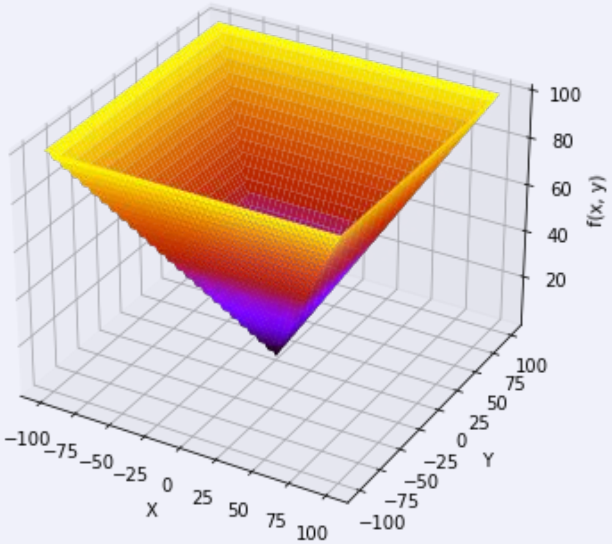


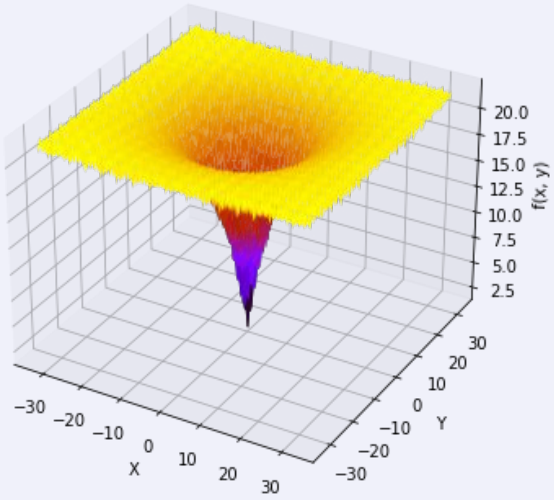
Alpine 1

$$f(\vec{x}) = \sum_{i=1}^{D} |x_i sin(x_i) + 0.1x_i|$$

Schwefel 2.21

$$f(\vec{x}) = \max_{i=1,\ldots,D} |x_i|$$



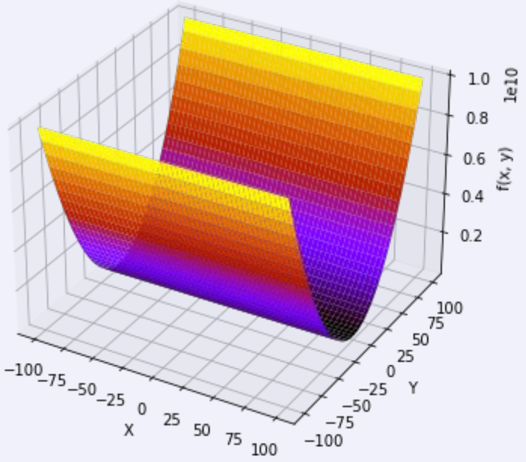


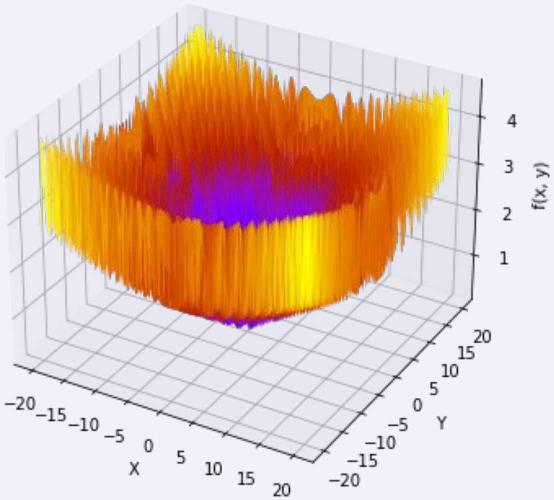
Ackley

$$f(\vec{x}) = -20 \cdot exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right) + e + 20$$

Bent Cigar

$$f(\vec{x}) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$$





Salomon

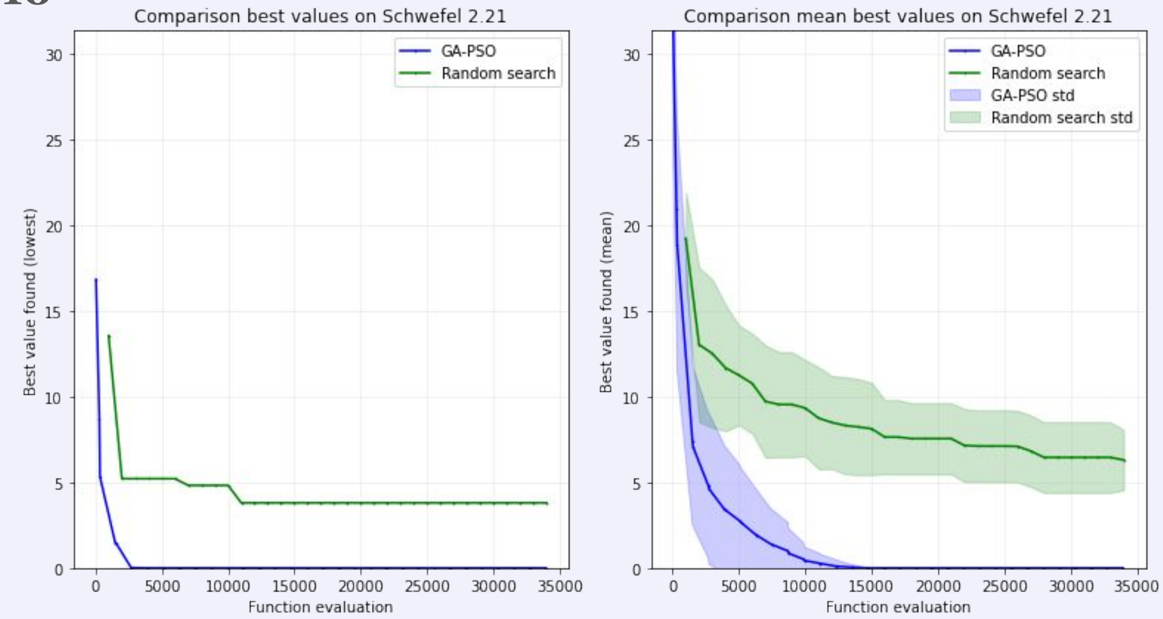
$$f(\vec{x}) = 1 - cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$$

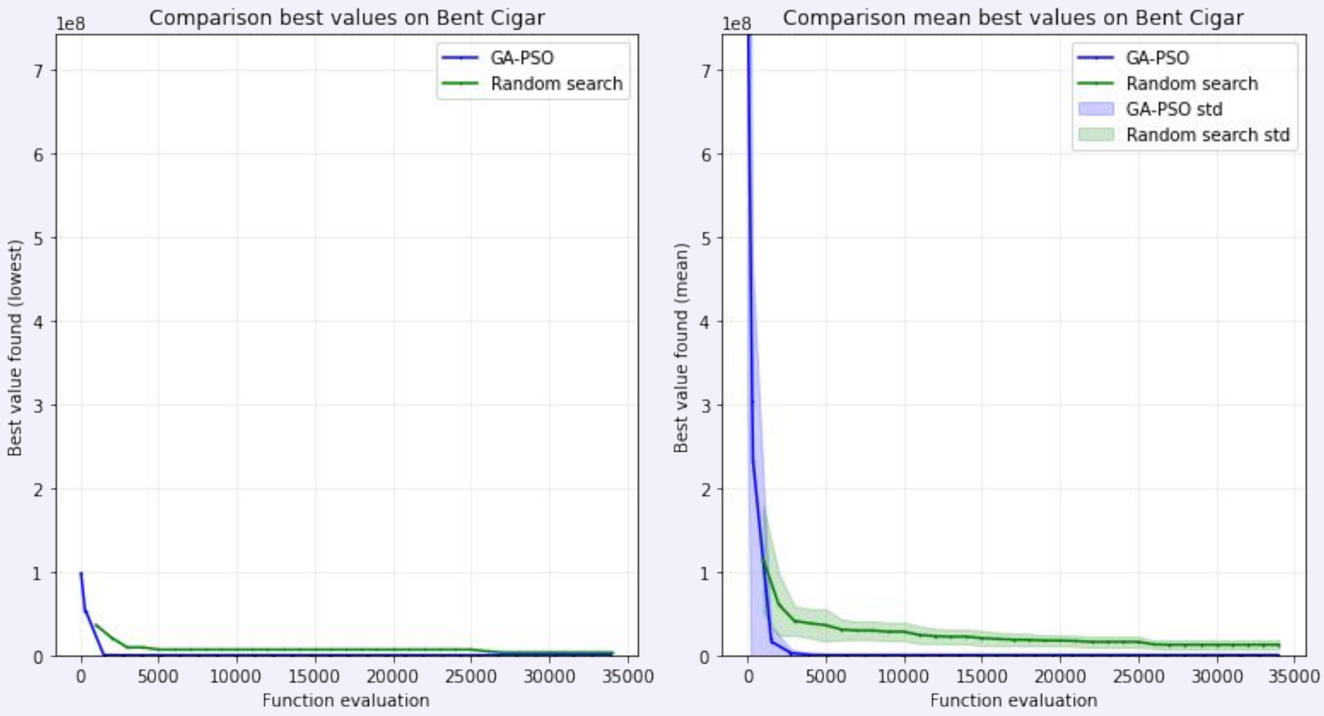
Some **low dimensional** results

- Population size: **50**
- Number of dimensions: **4** (2 **discrete** and 2 **continuous**)
- Number of discrete values: **101** (each coordinate)
- PSO iterations: **5**
- Number of independent runs: **10**

Schwefel 2.21

GA-PSO	$2.93 \cdot 10^{-50}$
RS	3.80





Bent Cigar

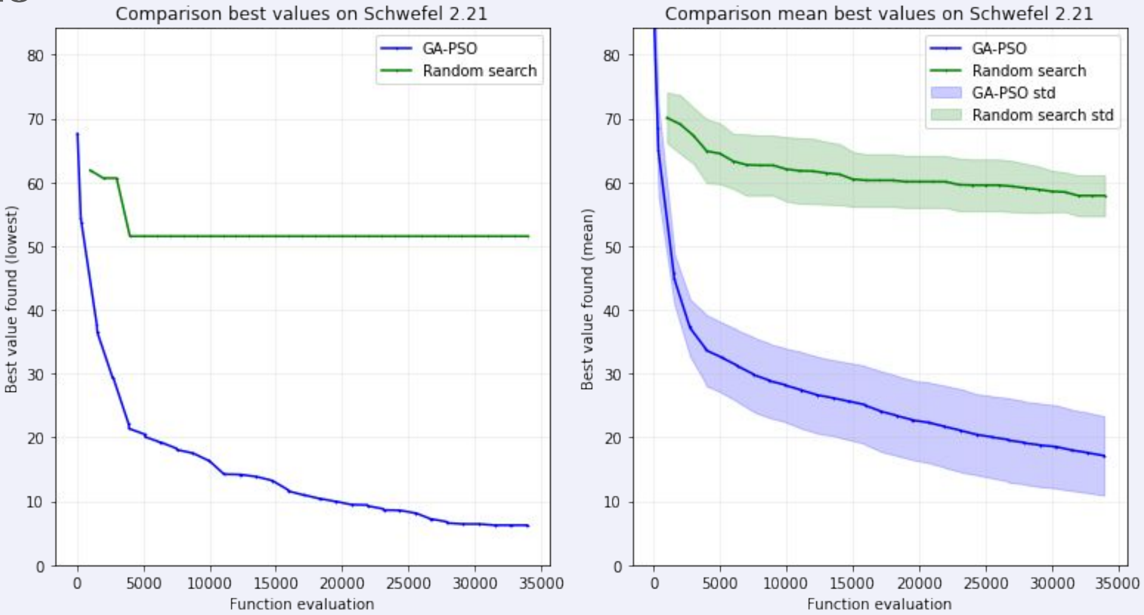
GA-PSO	$9.3 \cdot 10^3$
RS	$3.2 \cdot 10^6$

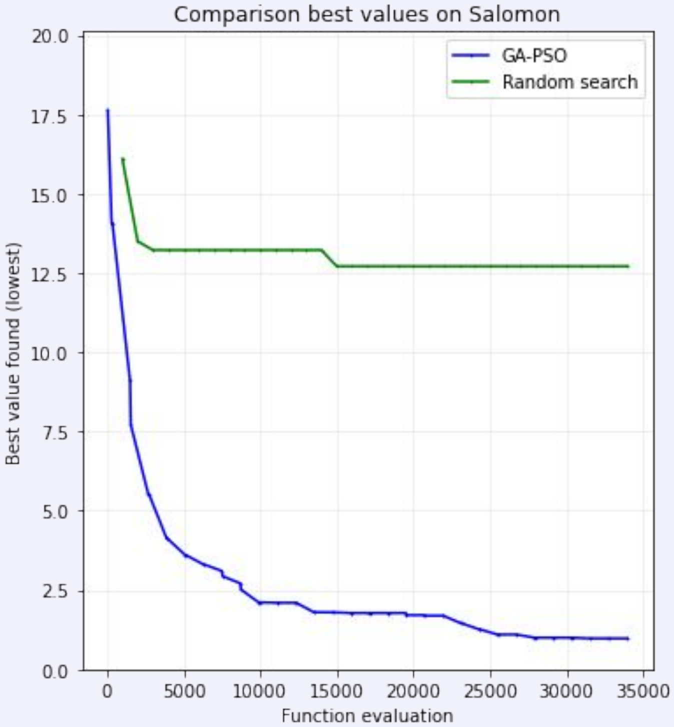
Some **intermediate dimensional** results

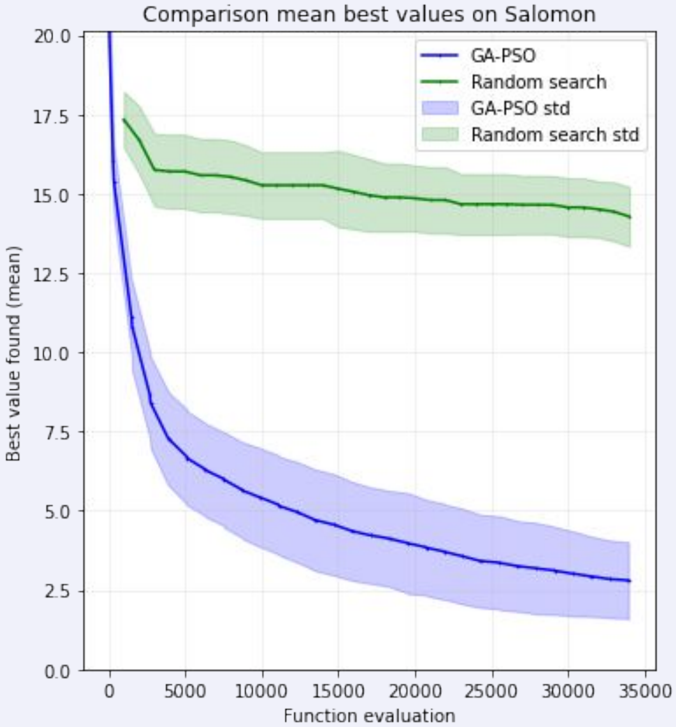
- Population size: **50**
- Number of dimensions: **20** (10 **discrete** and 10 **continuous**)
- Number of discrete values: **101** (each coordinate)
- PSO iterations: **5**
- Number of independent runs: **10**

Schwefel 2.21

GA-PSO	6.24
RS	51.5





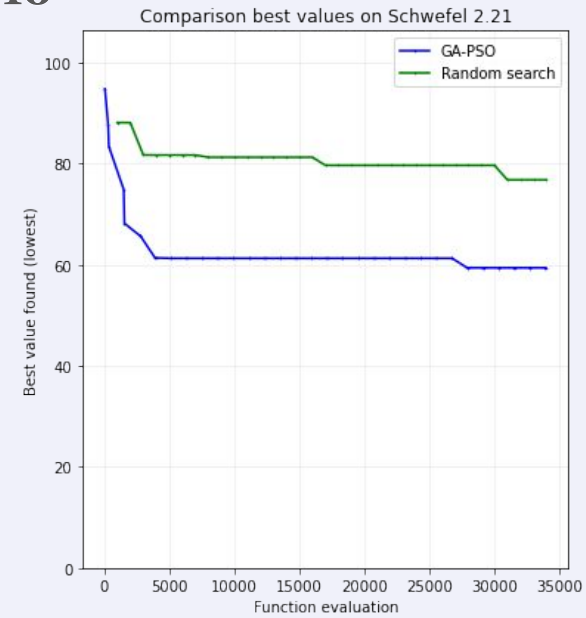


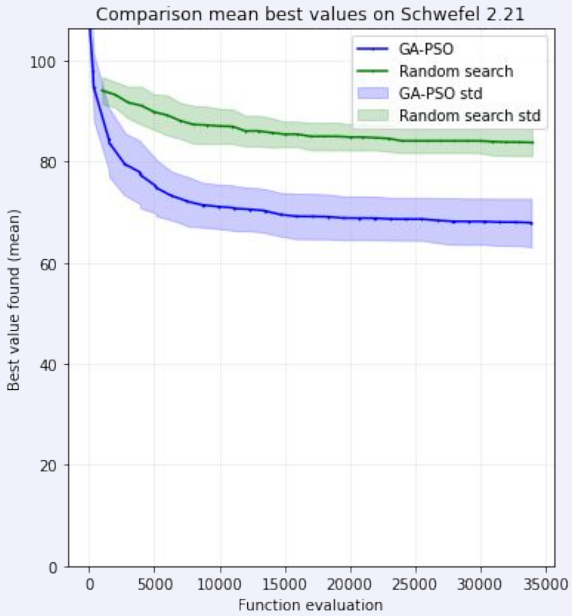
Salomon

GA-PSO	0.97
RS	12.7

Higher dimensional results

- Population size: **50**
- Number of dimensions: **50** (25 **discrete** and 25 **continuous**)
- Number of discrete values: **401** (each coordinate)
- PSO iterations: **5**
- Number of independent runs: **10**

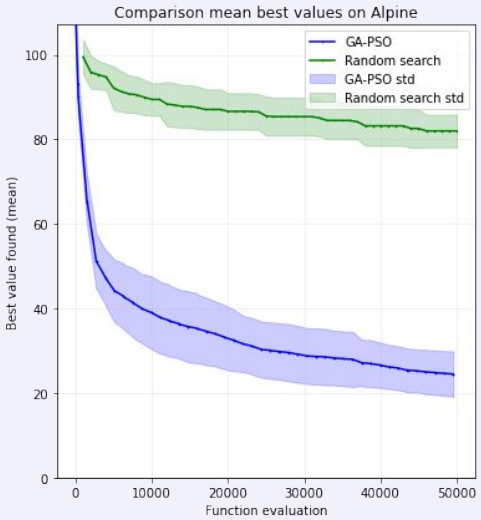




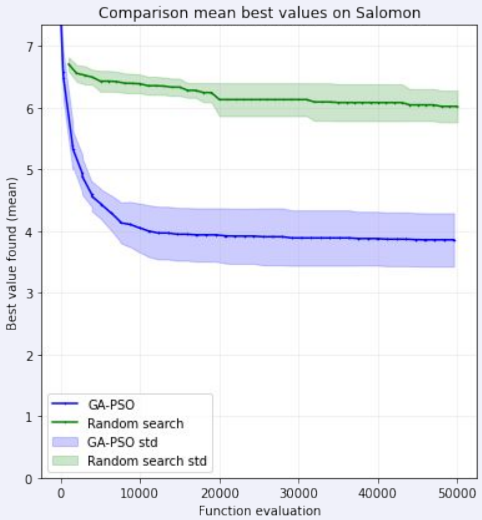
Schwefel 2.21

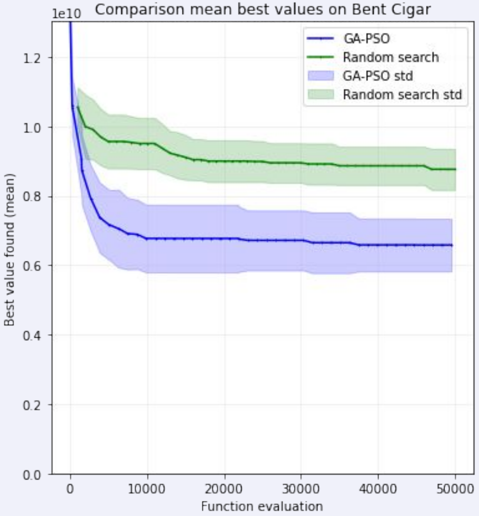
GA-PSO	59.4
RS	76.8

Alpine 1

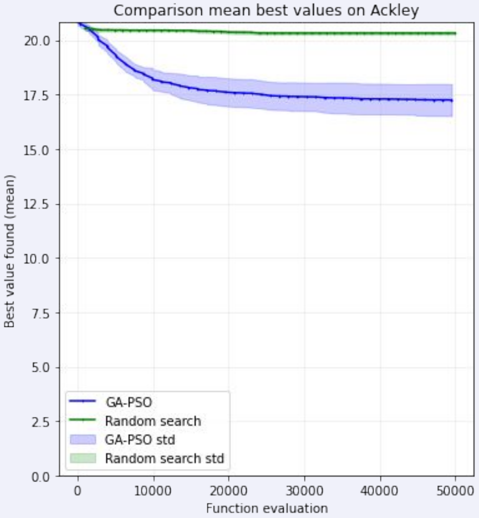


Salomon



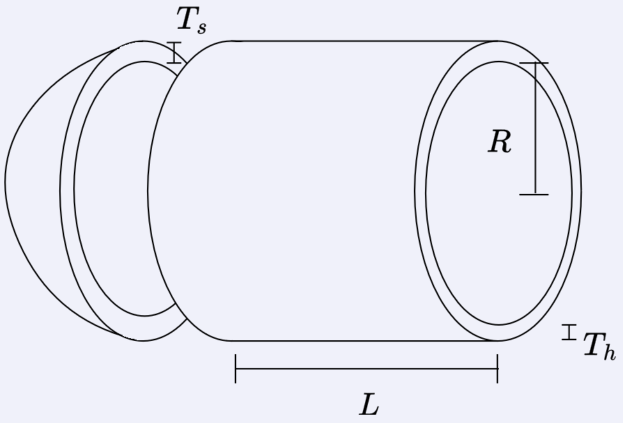


Bent Cigar



Ackley

Pressure Vessel Design optimization problem



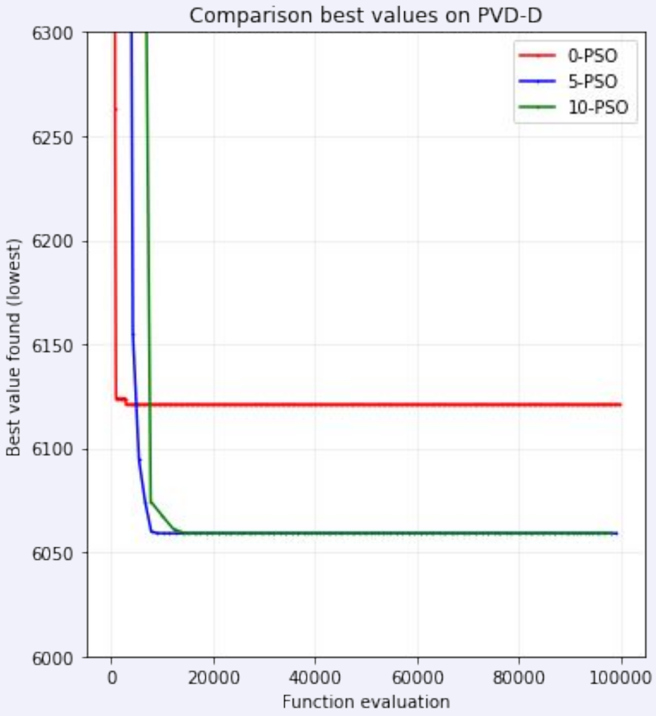
$$f(T_s, T_h, R, L) = 0.6224 \cdot T_s \cdot R \cdot L + 1.7781 \cdot T_h \cdot R^2 + 3.1611 \cdot T_s^2 \cdot L + 19.84 \cdot Ts^2 \cdot R$$

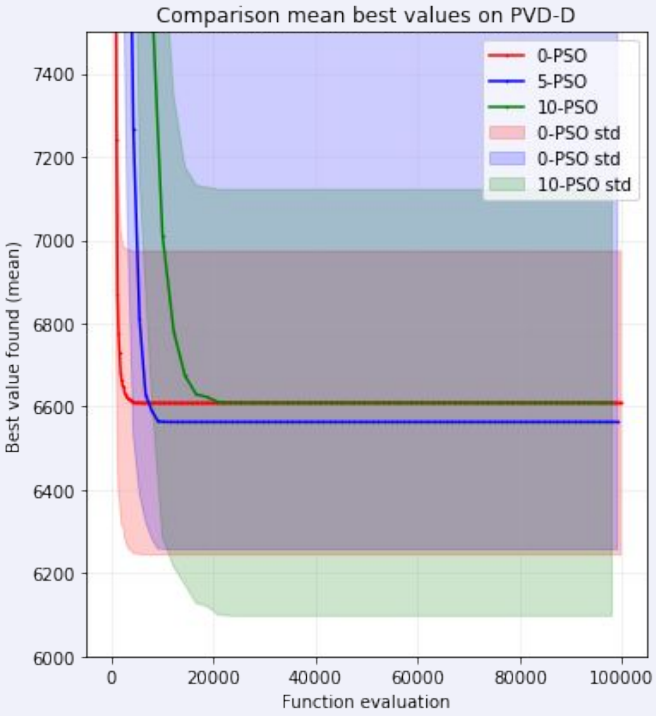
while satisfying the following **constraints**

$$R - 10 \geq 0$$
$$L - 200 \leq 0$$
$$T_s, T_h \in \{0.0625 \cdot i | i = 1, 2, \ldots, 99\}$$

$$-T_h + 0.0193R \leq 0$$
$$-T_s + 0.00954R \leq 0$$
$$-\pi R^2 L - \frac{4}{3}\pi R^3 + 1296000 \leq 0$$

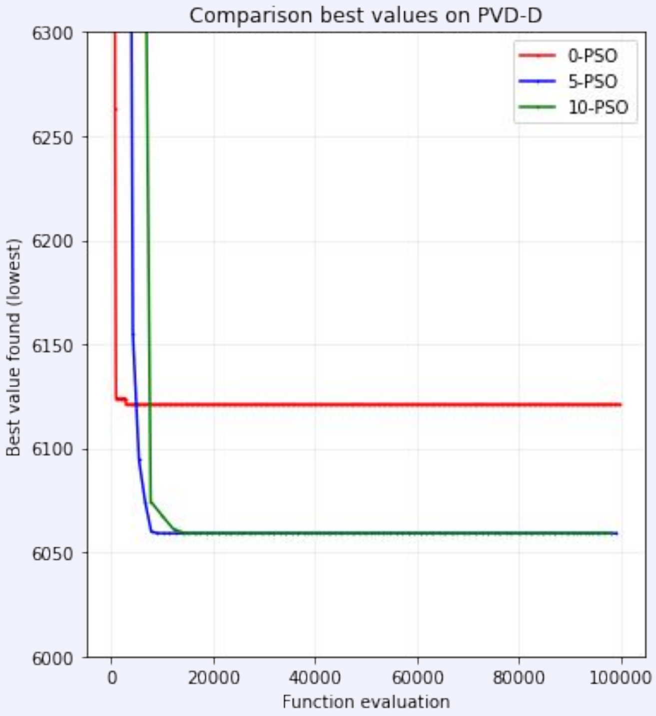
Constraints handling - **maintain**

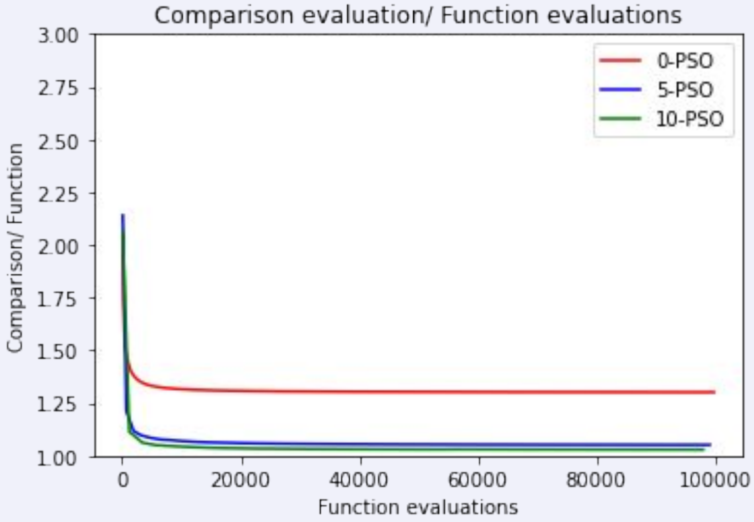




theoretical **minimum** is around 6059.714

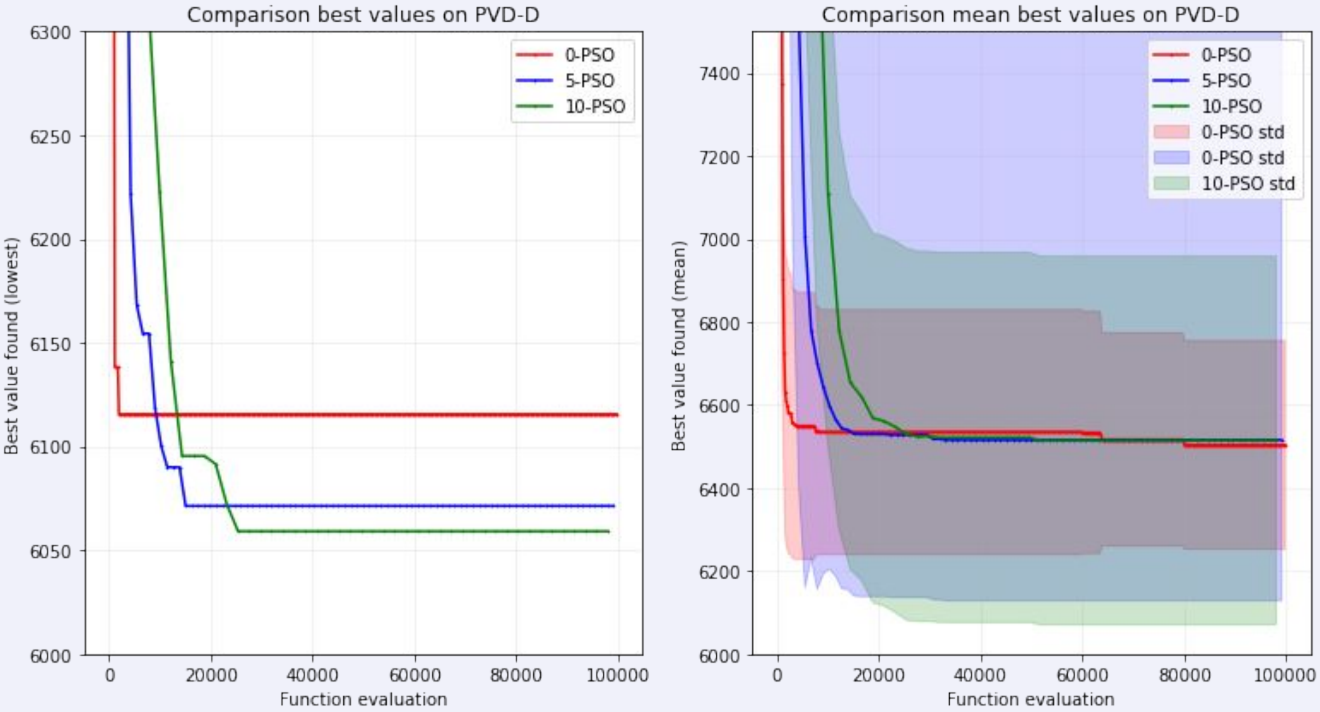
Constraints handling - **maintain**





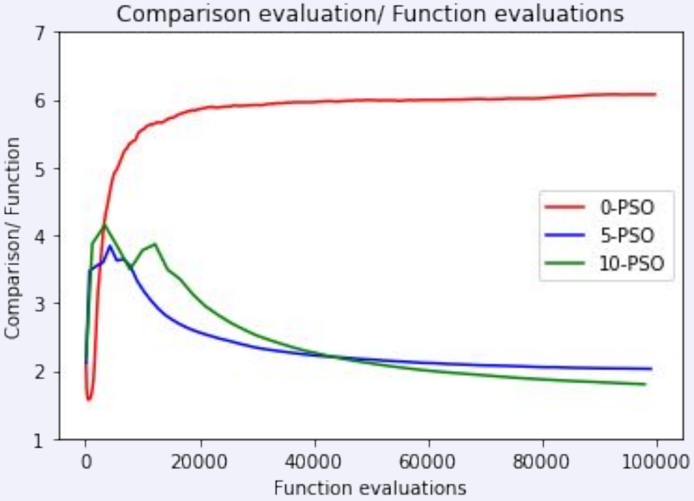
theoretical **minimum** is around 6059.714

Constraints handling - **repeat**



theoretical **minimum** is around 6059.714

Constraints handling - **repeat**



theoretical **minimum** is around 6059.714