



HAL
open science

Calibration & Optimisation

Sébastien Rey-Coyrehourcq, Etienne Delay, Paul Chapron

► **To cite this version:**

Sébastien Rey-Coyrehourcq, Etienne Delay, Paul Chapron. Calibration & Optimisation. École thématique. Calibration & Optimisation, Châtenay-sur-Seine, France. 2019. cel-02284550

HAL Id: cel-02284550

<https://hal.science/cel-02284550v1>

Submitted on 11 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Calibration & Optimisation

Etienne Delay
Sébastien Rey-Coyrehourcq
& Paul Chapron

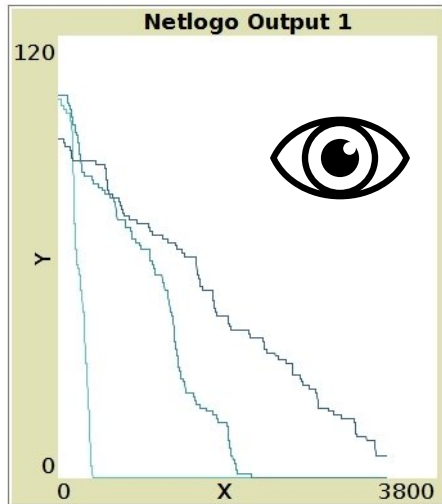
exModelo 2019



@reyman64
@ElCep

Model at the first sight

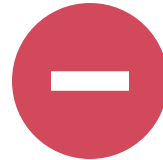
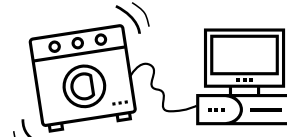
"One-Variable-At-A-Time"



"Face Validity"

Local sensitivity - OAT

- unreliable
- time-consuming
- limited
- difficult to reproduce



- playful
- interactive
- exploratory
- incremental
- trial error



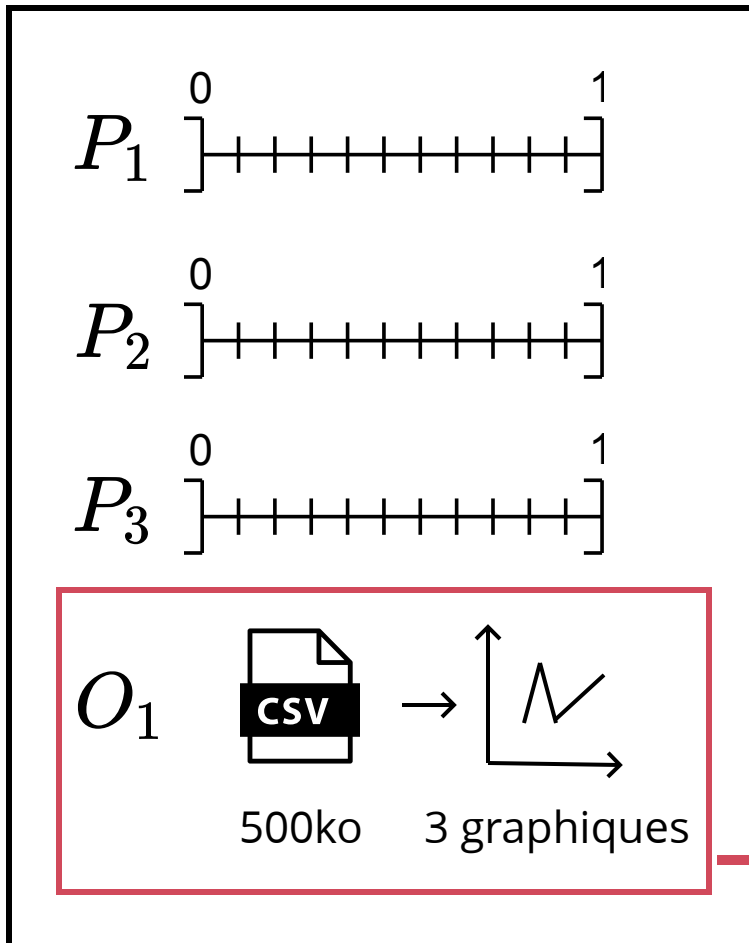
"**Tinkering**" Papert & Retsnick

How? How?



A first step to systematize OAT?

=> Overall Sensitivity



a simple model,
a complete experimental design:

P1 x P2 x P3

systematic outputs

How? How?



Issue no. 1 - Combinatorial explosion

$$\text{discreteStep}^{\left(\sum_{i=1}^n P_i\right)} * \text{replication} * \text{time}$$

$$\{\text{replication} \in \mathbb{R} \mid 30\}$$

$$\{\text{duree} \in \mathbb{R} \mid 1\text{minute}\}$$

$$\text{discreteStep} = 11 \quad P = 3$$

$$11^3 * 30 * 1\text{min} = 39930\text{min} = 27\text{days}$$

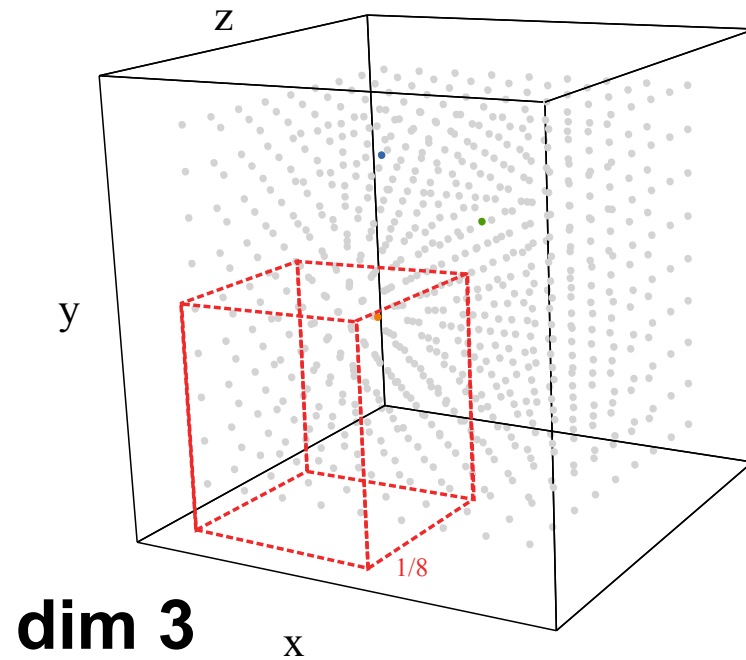
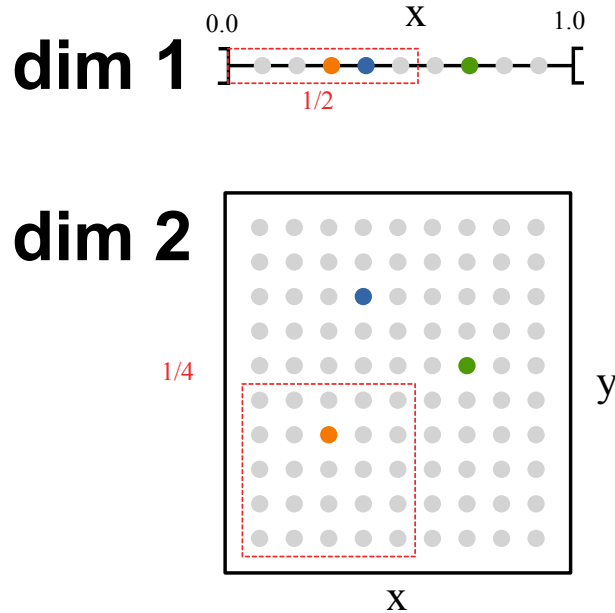
$$11^3 * 30 * 500\text{ko} = 19965000\text{Ko} = 19\text{Go}$$

$$11^3 * 30 * 3 \text{ graph} = 119790 \text{ graphs}$$

Duration & Volumetry impossible to manage

How? How?

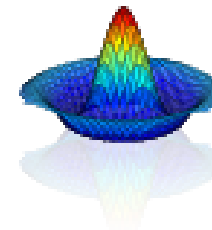
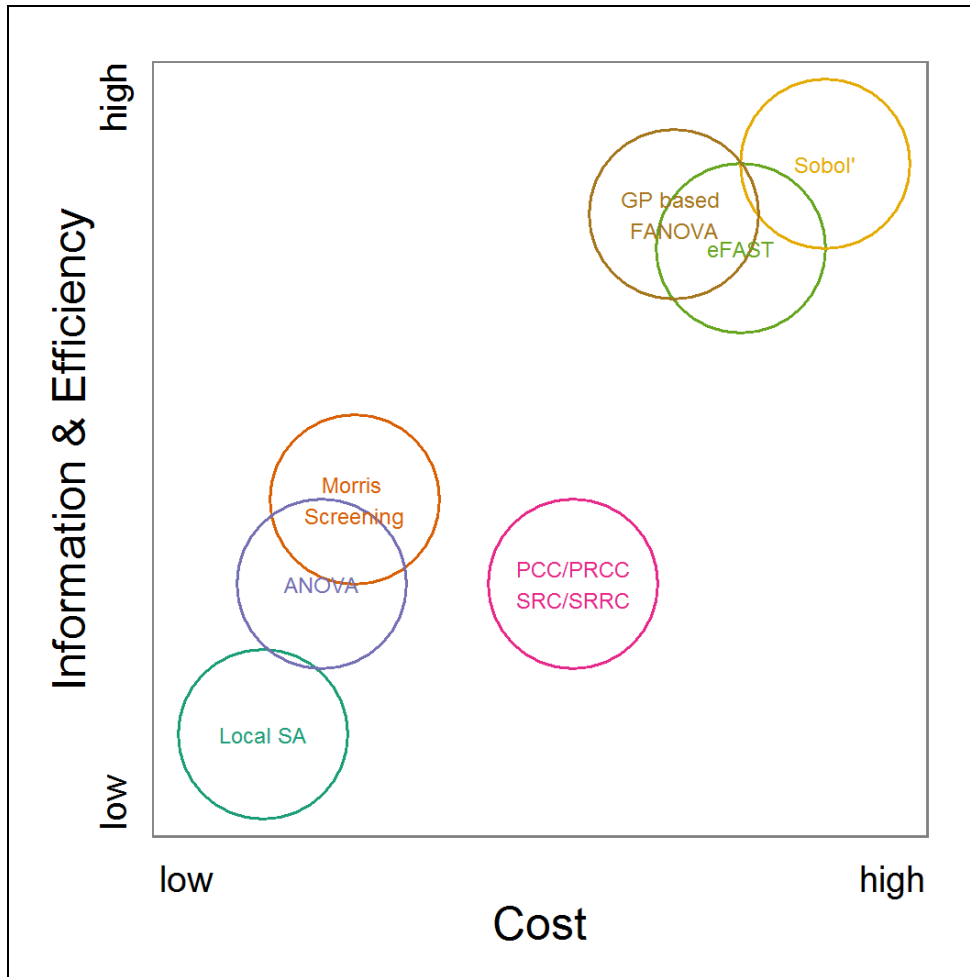
Issue no. 2 - "Curse Dimensionality"



+ dimensions = **-** coverage at equal nb points

How? How?

A first solution, more sophisticated methods.

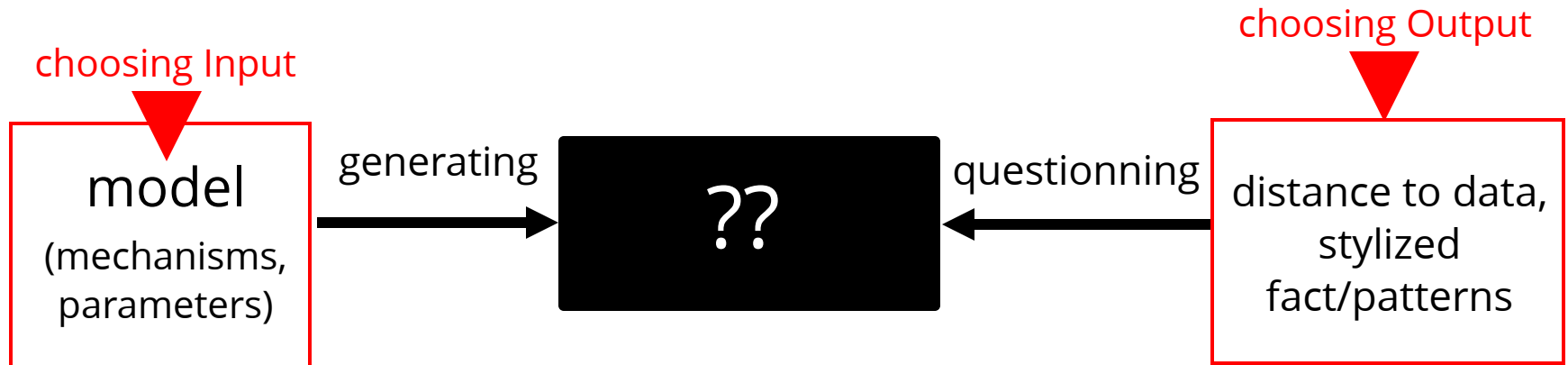


MEXICO
WEXICO

<http://reseau-mexico.fr/>

How? How?

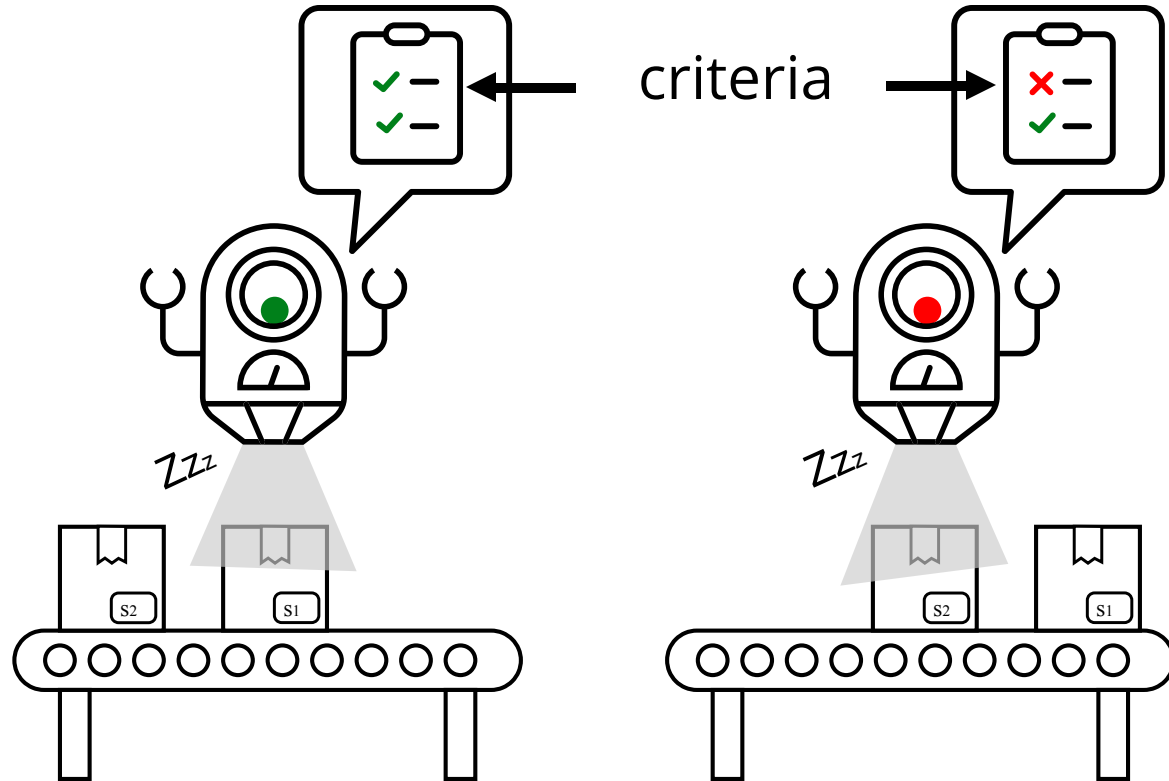
A second solution, change your approach!



Organize a **tension** between the **Input** choices and **Output** choices

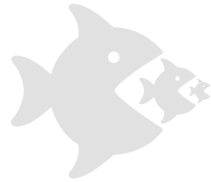
How? How?

What tensioning?



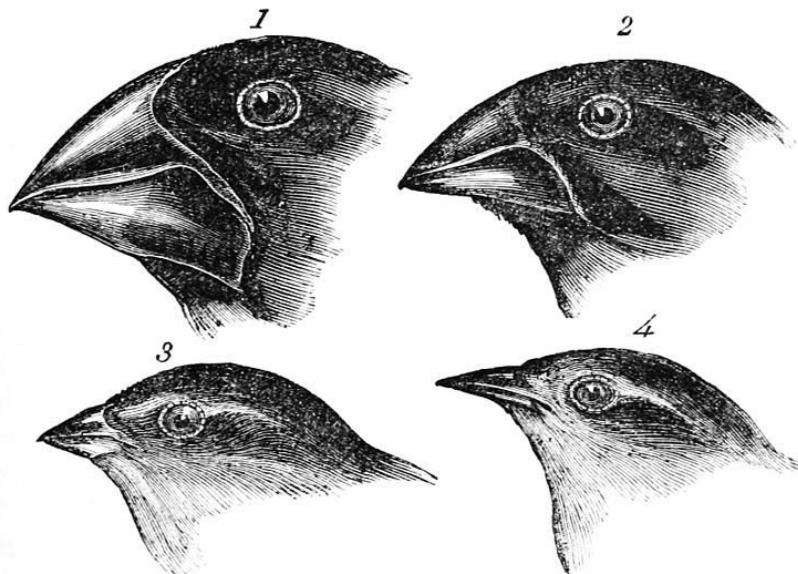
Quantified criteria to "guide"
automated exploration

Evolutionary Algorithm



Darwin like algorithms

Evolutionary algorithms (EAs) are population-based metaheuristic optimization algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection, and **survival of the fittest** in order to refine a set of solution candidates iteratively.



1. *Geospiza magnirostris*.
3. *Geospiza parvula*.

2. *Geospiza fortis*.
4. *Certhidea olivacea*.

Advantages :

- Population based => perfect for HPC
- As good for **Exploration** than **Exploitation**
- Optimize one or multiples objectives

Inconvenients :

- Greedy !
- Black Box
- Lot of parameters
- Termination criteria ?

Definitions

Objectives vs Fitness/Heuristic

Heuristic function

is a way to inform the search about the direction to a goal. **It provides an informed way to guess which path is promising.** The heuristic function can be **direct or indirect measurement**, for example by providing only an approximation of the distance between a measurement and the optimum.

Here's an **algorithm** for driving to someone's house :

```
Take Highway's 167 south to Puyallup. Take the South Hill Mall exit and drive 4.5 miles up the hill. Turn into the drive away of the large tan house on the left, at 714 North Cedar
```

Here's an **heuristic** for getting to someone's house :

```
Find the last letter we mailed you. Drive to the town in the return address. When you get to town, ask someone where our house is. If you can't find anyone, call us from a public phone, and we'll come get you.
```

based on [Mac Connell 2004]

Definitions

Objectives vs Fitness/Heuristic

Objective function

The objective function can be considered as a form of heuristics, except that it is a **direct measure** of the potential of an aspect of the solution. In this sense, the objective function often requires more expertise on the system than the heuristics.

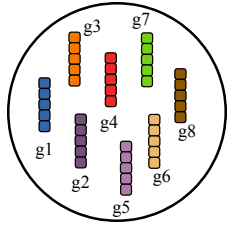
based on [Weise2011 p36]

Fitness function

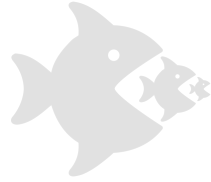
A fitness function is a particular type of objective function that is used to summarise, as a single figure of merit, how close a given design solution is to achieving the set aims.

based on [Wikipedia](#)

Evolutionary Algorithm

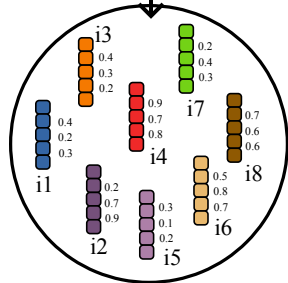
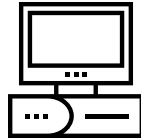


$$P = G_0..G_n$$



Translation to simulation

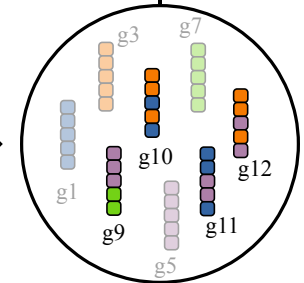
simulation



$$I_1 = G_1 + \{O_0...O_m\}$$

...

$$I_n = G_n + \{O_0...O_m\}$$



mutate & crossover
values

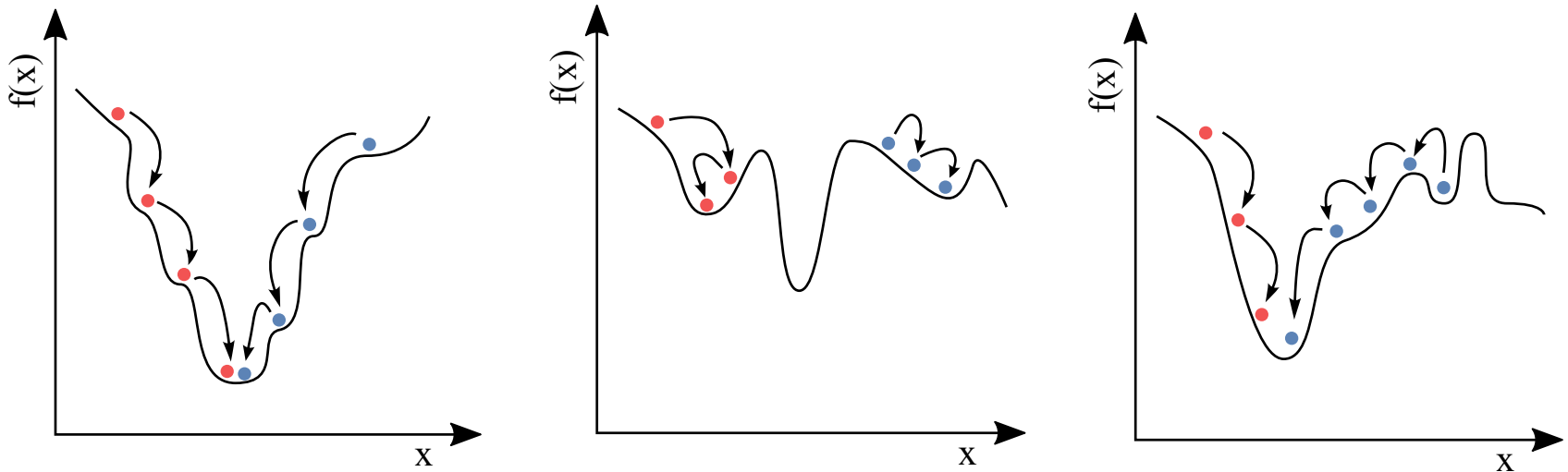
Genome contain
values of parameters

Evolutionary Algorithm



Importance of both
Exploration & Exploitation

Some fitness landscape to understand (1 obj.)



easy landscape,
one global
optima



blocked on local
optima

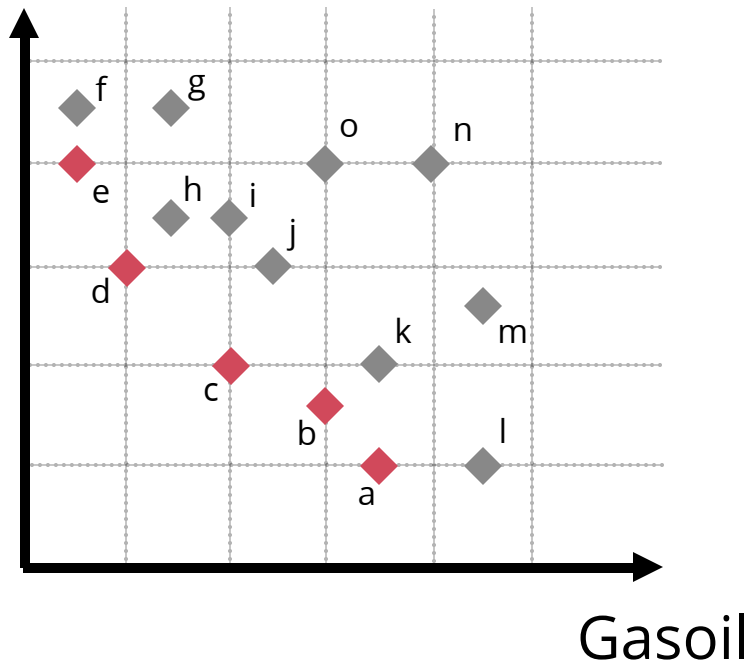


jump outside
local optima

Take the best and try again ?

Objective space : Pareto introduction

Price



Each car ◆ dominates all others
by at least one objective

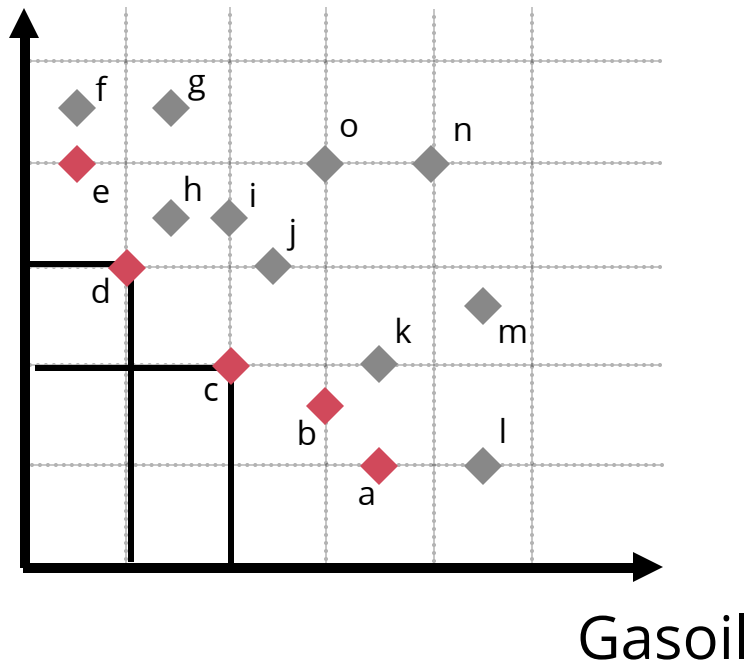
- ➔ Non-Dominated Front
- ➔ Pareto front

Vilfredo Pareto Economist 1950

Take the best and try again ?

Objective space : Pareto introduction

Price



Each car ◆ dominates all others by at least one objective

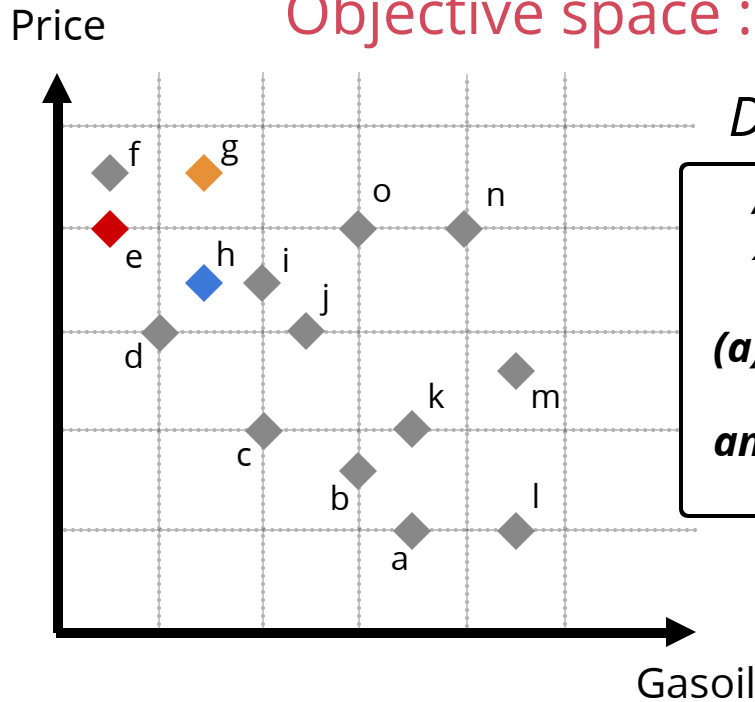
- ➔ Non-Dominated Front
- ➔ Pareto front

| | | |
|--------|----------|----------|
| | d | c |
| Price | 3 | 2 |
| Gasoil | 1 | 2 |

Vilfredo Pareto Economist 1950

Take the best and try again ?

Objective space : **Dominance** based criteria



Definition :

An element x_1 dominates (is preferred to) an element x_2 ($x_1 \dashv x_2$) if:

- (a)** x_1 is **better** than x_2 in at least one objective function
- and (b)** not worse with respect to all other objectives.

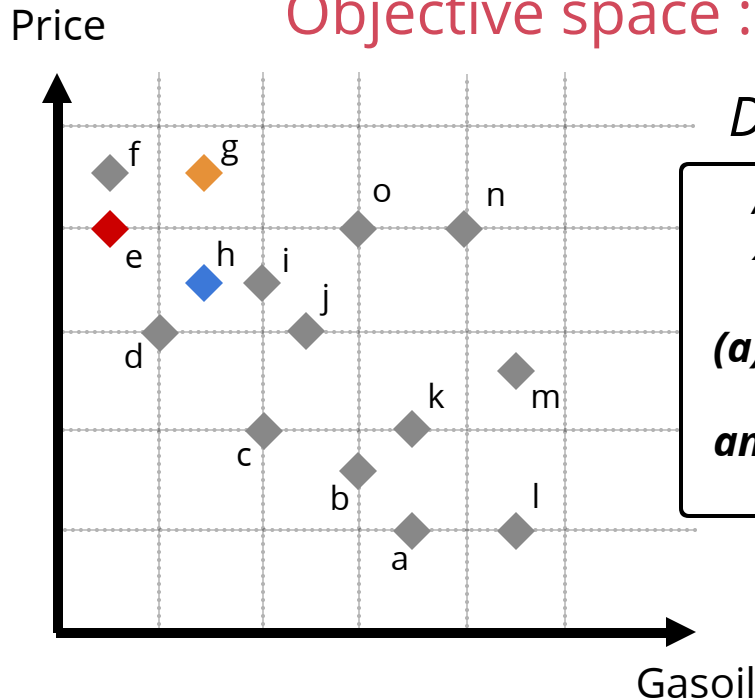
$$(x_1 \dashv x_2) \Leftrightarrow \forall i \in 1 \dots n \Rightarrow f_i(x_1) \leq f_i(x_2) \wedge \exists j \in 1 \dots n : f_j(x_1) < f_j(x_2)$$

e dominate h ?

e dominate g ?

Take the best and try again ?

Objective space : **Dominance** based criteria



Definition :

An element x_1 dominates (is preferred to) an element x_2 ($x_1 \dashv x_2$) if:

- (a)** x_1 is **better** than x_2 in at least one objective function
- and (b)** not worse with respect to all other objectives.

$$(x_1 \dashv x_2) \Leftrightarrow \forall i \in 1 \dots n \Rightarrow f_i(x_1) \leq f_i(x_2) \wedge \exists j \in 1 \dots n : f_j(x_1) < f_j(x_2)$$

e dominate h ?

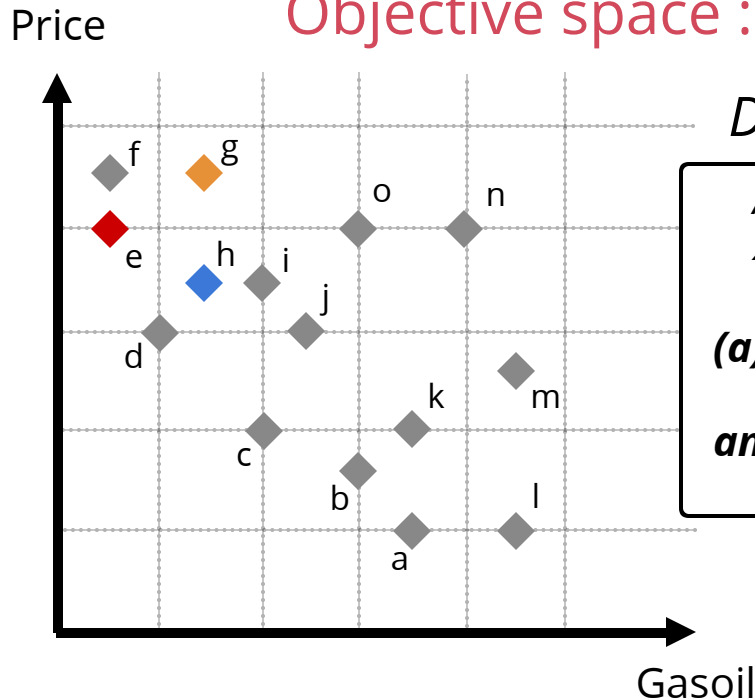
*e better on Price (False) ? Gasoil (True) ?
e not worse on Price (False) ?*

➔ *e doesn't dominate h*

e dominate g ?

Take the best and try again ?

Objective space : **Dominance** based criteria



Definition :

An element x_1 dominates (is preferred to) an element x_2 ($x_1 \dashv x_2$) if:

- (a)** x_1 is **better** than x_2 in at least one objective function
- and (b)** not worse with respect to all other objectives.

$$(x_1 \dashv x_2) \Leftrightarrow \forall i \in 1 \dots n \Rightarrow f_i(x_1) \leq f_i(x_2) \wedge \exists j \in 1 \dots n : f_j(x_1) < f_j(x_2)$$

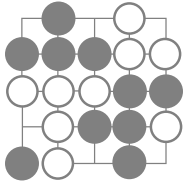
e dominate h ?

*e better on Price (False) ? Gasoil (True) ?
e not worse on Price (False) ?*

➔ *e doesn't dominate h*

e dominate g ?

e better on Price (True) ? Gasoil (True) ?

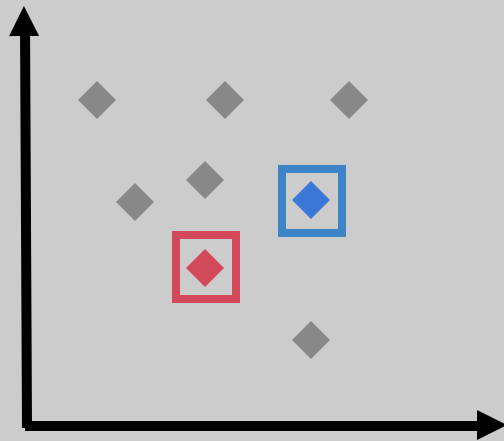


Not so easy !

Maintaining a **diversity** of solution

Like many board games... winning n battles doesn't mean you win the whole war !

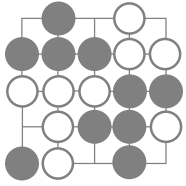
Battles lost



offsprings ●

War lost

Iteration 1

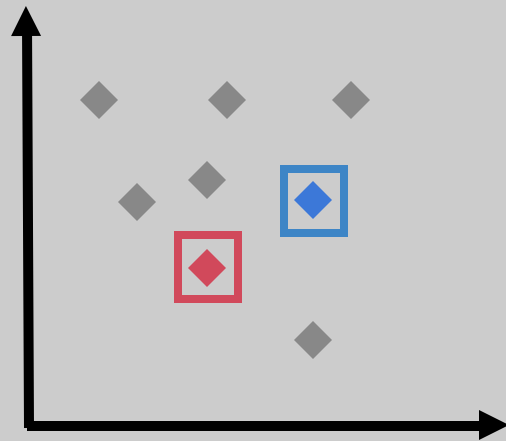


Not so easy !

Maintaining a **diversity** of solution

Like many board games... winning n battles doesn't mean you win the whole war !

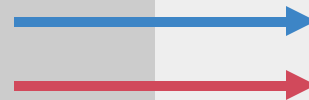
Battles lost



War lost

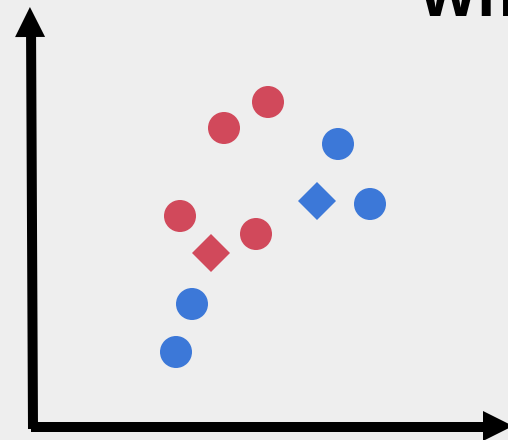
Iteration 1

offsprings ●



Battles lost

Why ?



War lost

Iteration 2
(only ♦ & ◆) childs

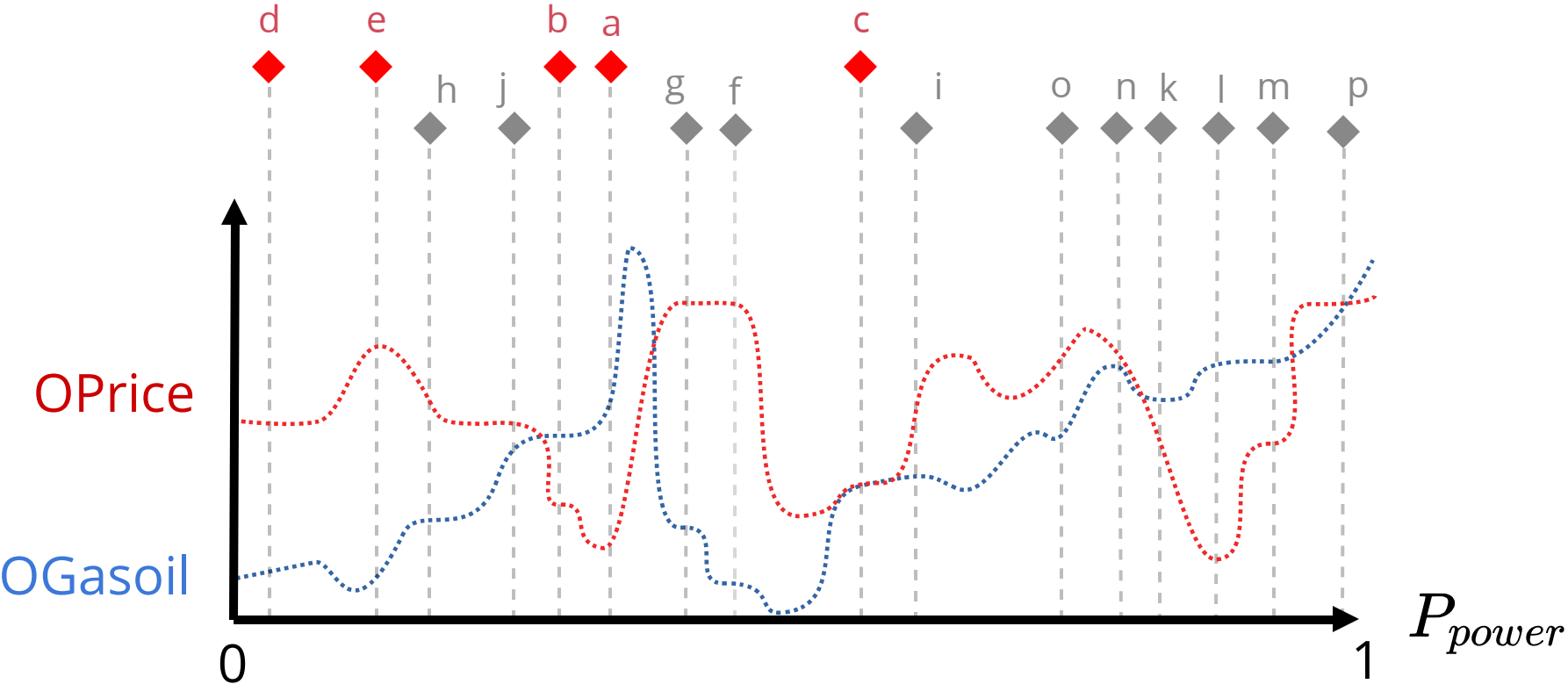
Imagine



Simulation with thousand parameter fixed for motors,
you play with only one : **power**.

We continue to search to minize **Price & Gasoil**
on cars.

..... **unknown laws** from optimizer ◆ pareto



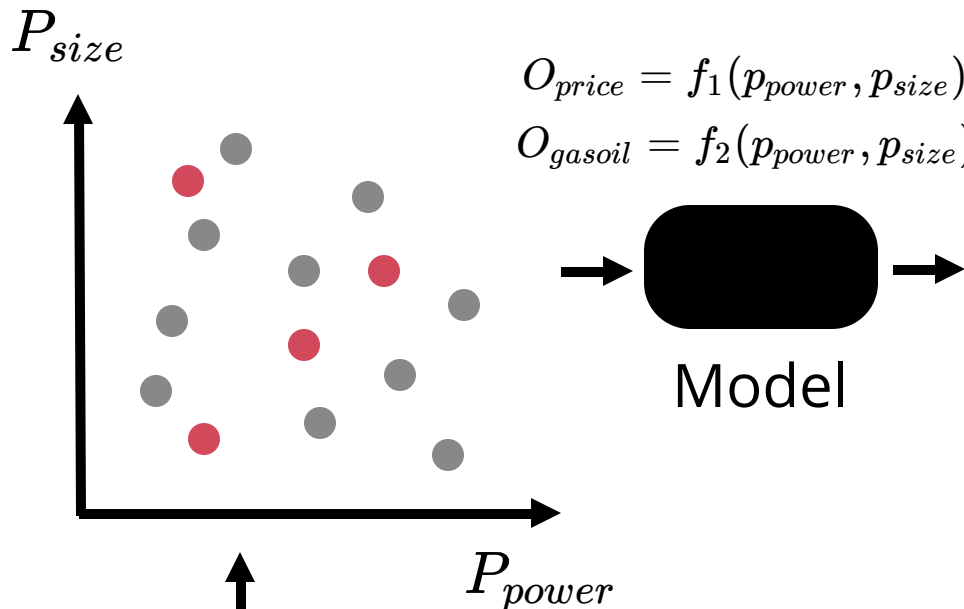
How to distinguish between overall good solutions and locally fit solutions

➡ optimizer [?] doesn't know !

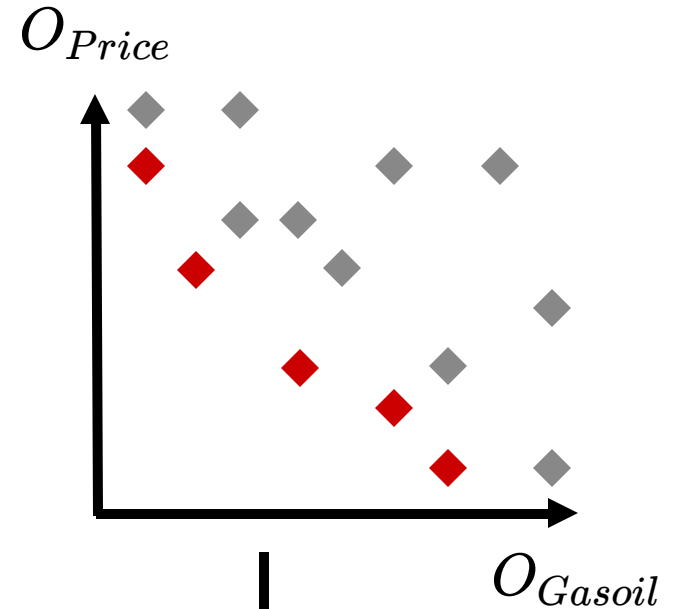
Not so easy !

Relations between parameters & objectives space

Parameter Space (2 param.)



Objective Space (2 obj.)



Optimizer : How i **select** & **move** on the next step ?

=> **Fitness** help to solve this

Take the best and try again ?

Resuming the situation

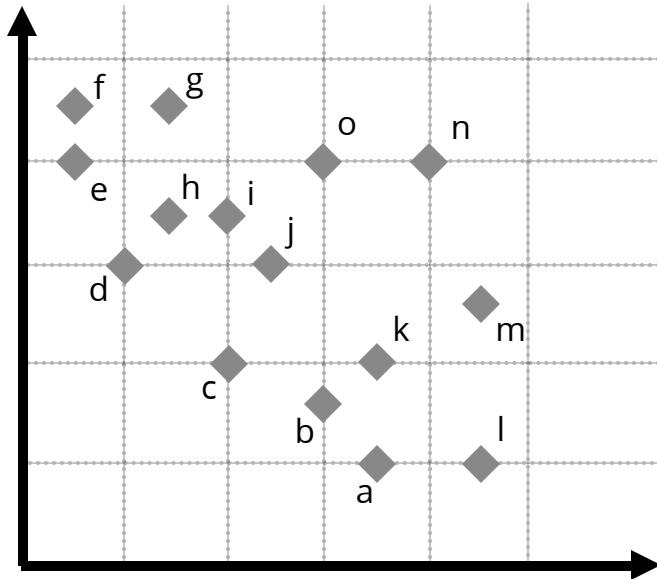


EA Algorithms proceed by **indirect move** on the **n dimensions** of Parameters space to get **better** values on Objectives space => FITNESS

Hyp. of EA : A little move on parameter space could produce a better result on obj. space ... (an it works really well in many cases ...)

Ranking ?

Pareto ranking & diversity algorithm

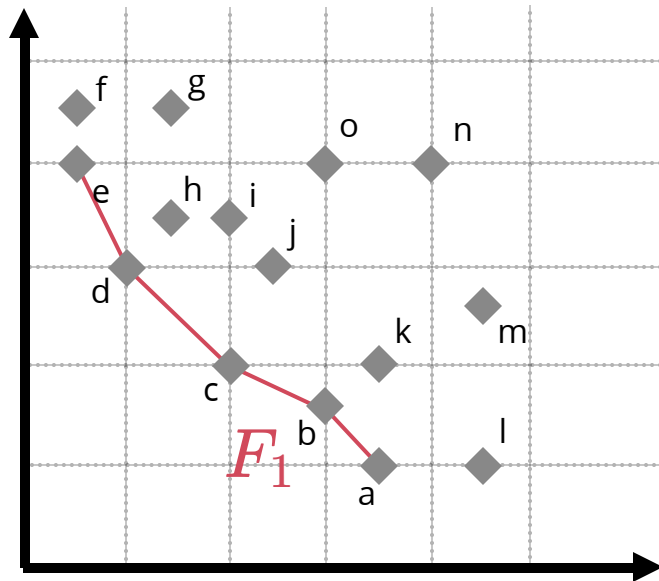


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

Ranking ?

Pareto ranking & diversity algorithm

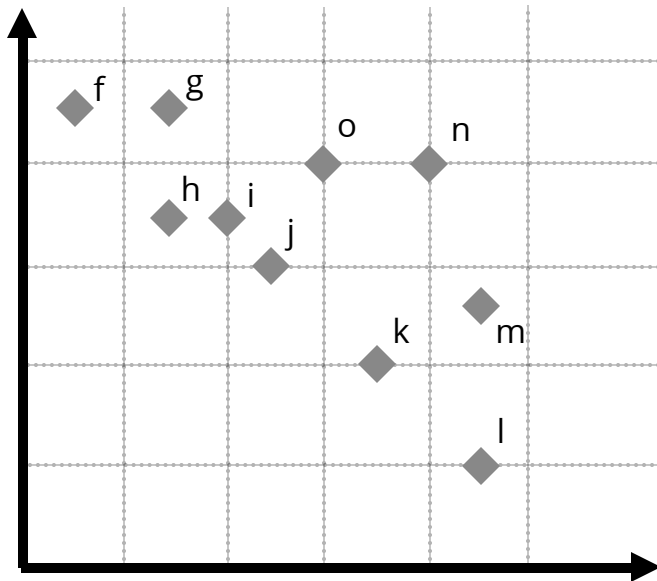


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

Ranking ?

Pareto ranking & diversity algorithm

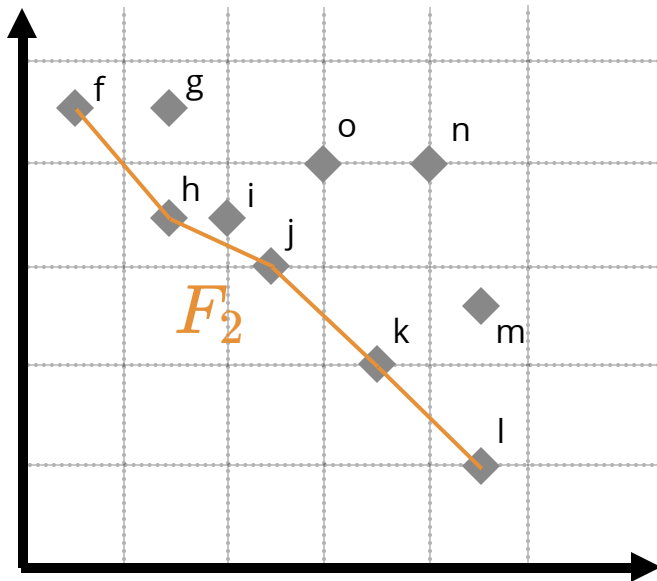


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

Ranking ?

Pareto ranking & diversity algorithm

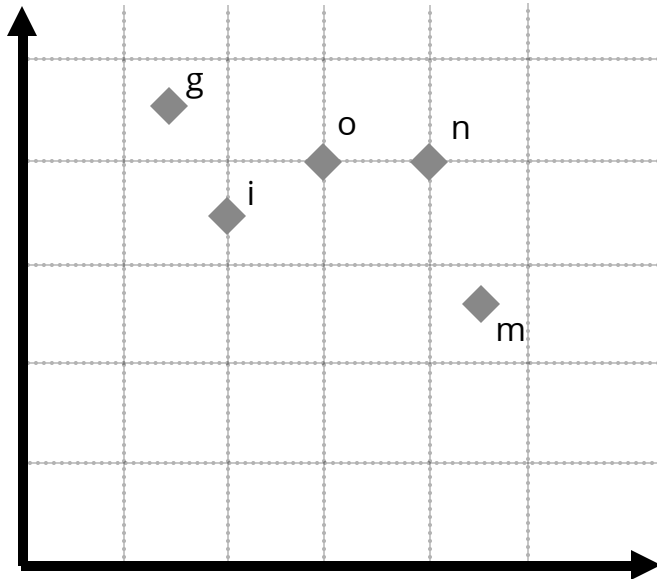


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

Ranking ?

Pareto ranking & diversity algorithm

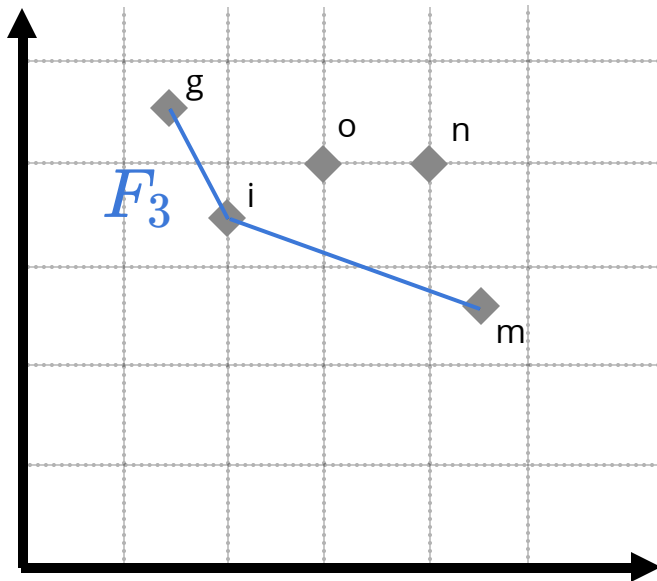


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

Ranking ?

Pareto ranking & diversity algorithm

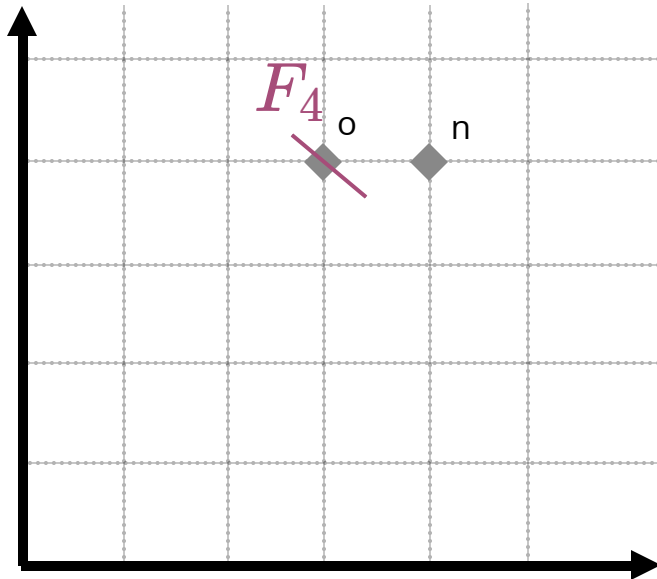


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

Ranking ?

Pareto ranking & diversity algorithm

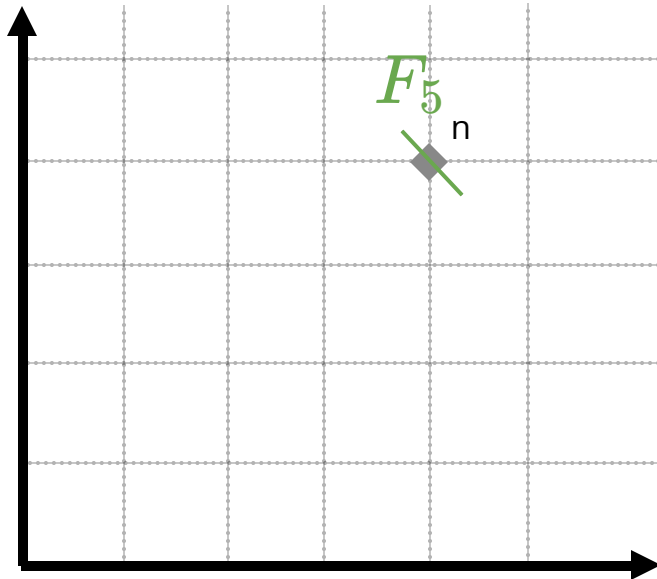


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

Ranking ?

Pareto ranking & diversity algorithm

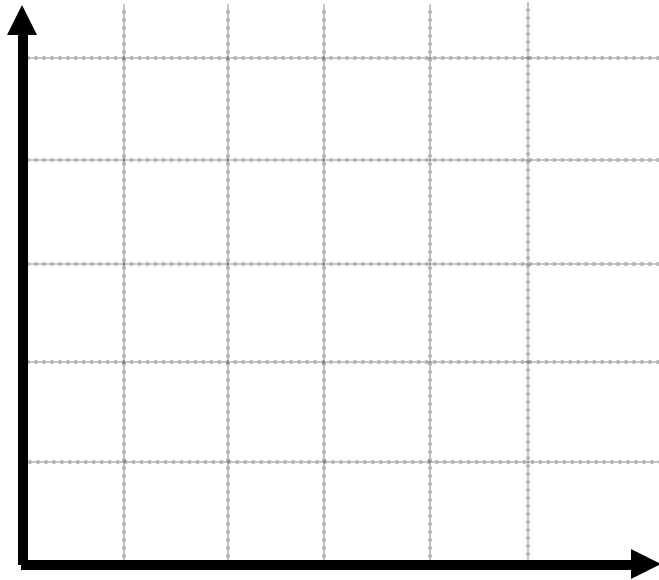


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

Ranking ?

Pareto ranking & diversity algorithm

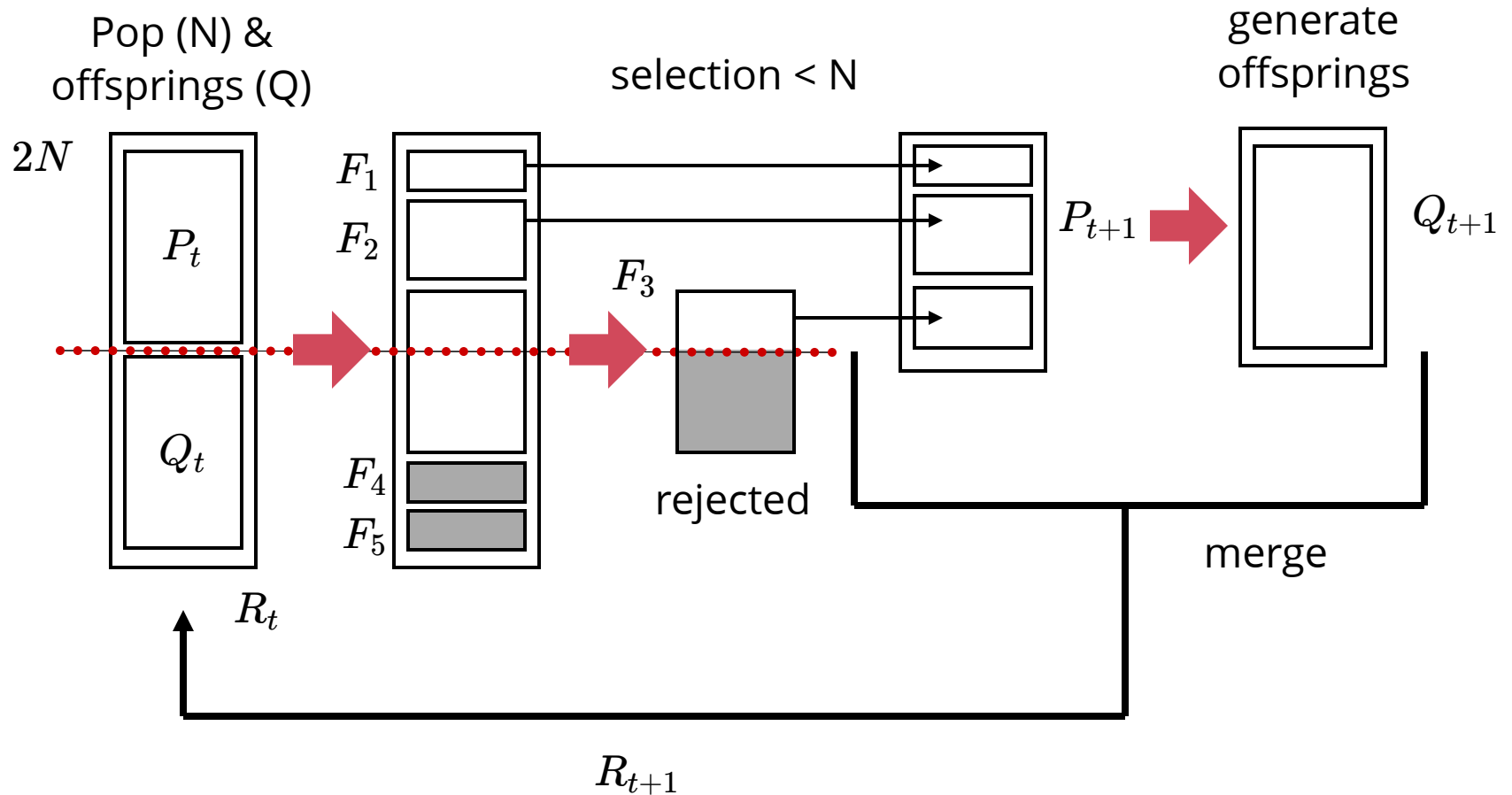


ranking is used to compute
a **fitness**

| | <i>f1</i> | <i>f2</i> | dominated by | r1 | r2 |
|---|-----------|-----------|--------------------------------|----|----|
| a | 3,5 | 1 | \emptyset | 1 | 1 |
| b | 3 | 1,5 | \emptyset | 1 | 1 |
| c | 2 | 2 | \emptyset | 1 | 1 |
| d | 0,5 | 4 | \emptyset | 1 | 1 |
| e | 0,5 | 4,5 | \emptyset | 1 | 1 |
| f | | | {e} | 2 | 2 |
| g | | | {d, e, f, h} | 3 | 5 |
| h | | | {d} | 2 | 2 |
| i | | | {c, d, h} | 3 | 4 |
| j | | | {c, d} | 2 | 3 |
| k | | | {a, b, c} | 2 | 4 |
| l | | | {a} | 2 | 2 |
| m | | | {a, b, c, k, l} | 3 | 6 |
| n | | | {a, b, c, d, e, h, i, j, k, o} | 5 | 11 |
| o | | | {b, c, d, e, h, i, j} | 4 | 8 |

NSGA 2

One famous algorithm

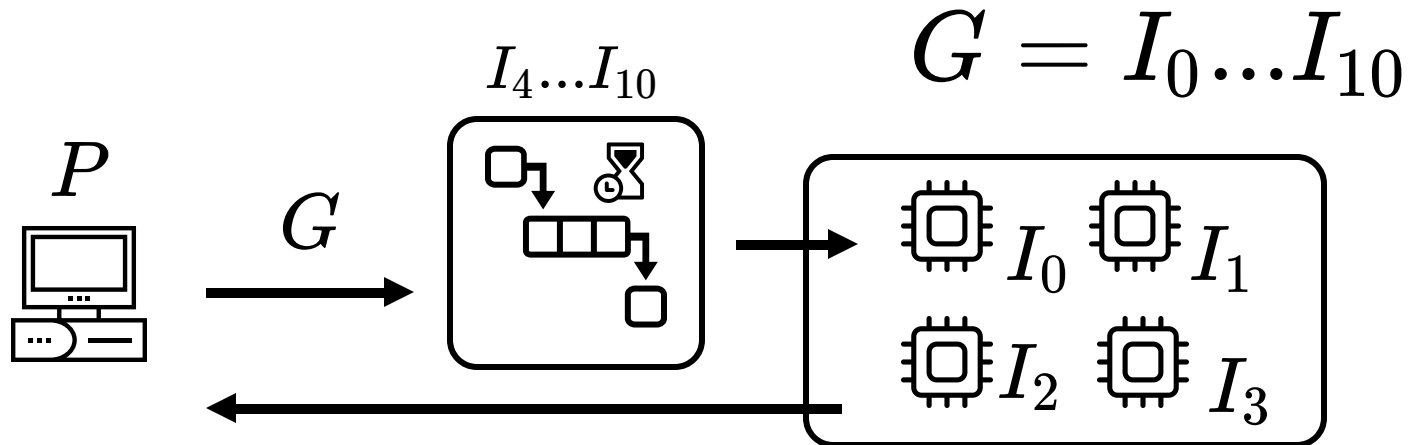


keeping the P_t candidates in the 3 first Pareto front to maintain diversity

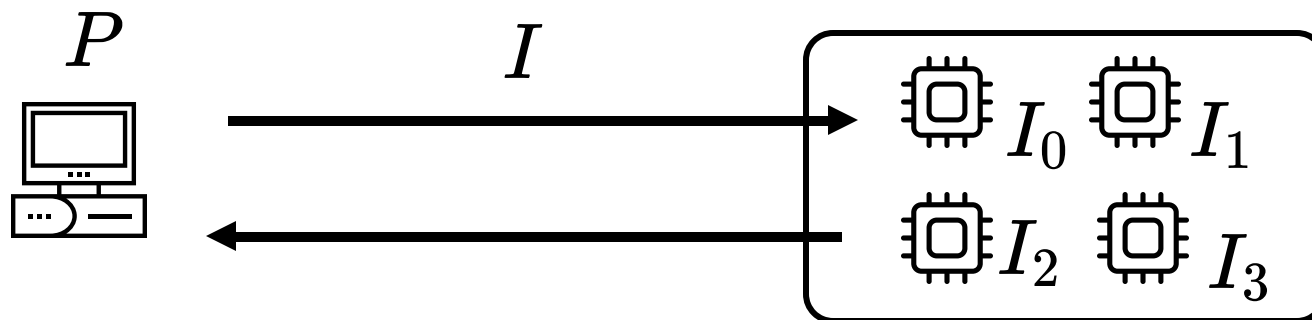
NSGA 2

OpenMole & MGO library

generational EA



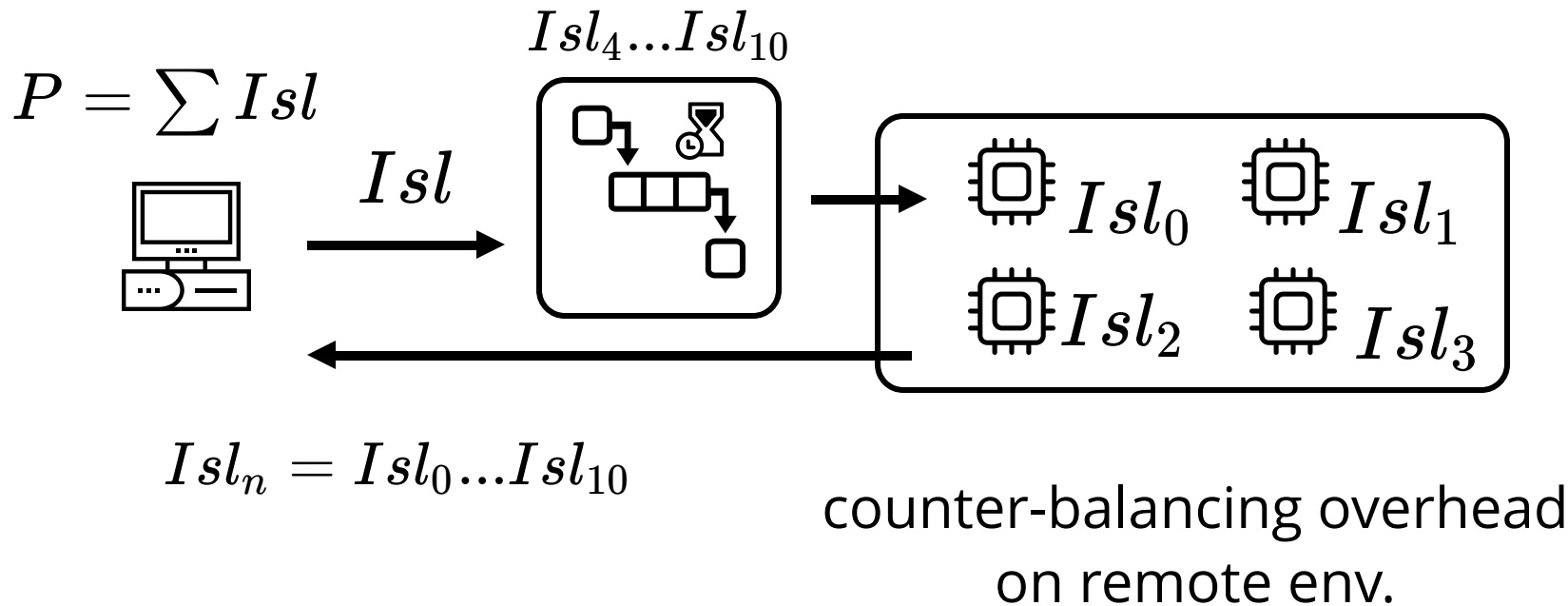
Steady State EA : *as soon as possible strategy*



NSGA 2

OpenMole & MGO library

Steady State Island EA



NSGA 2

OpenMole Syntax

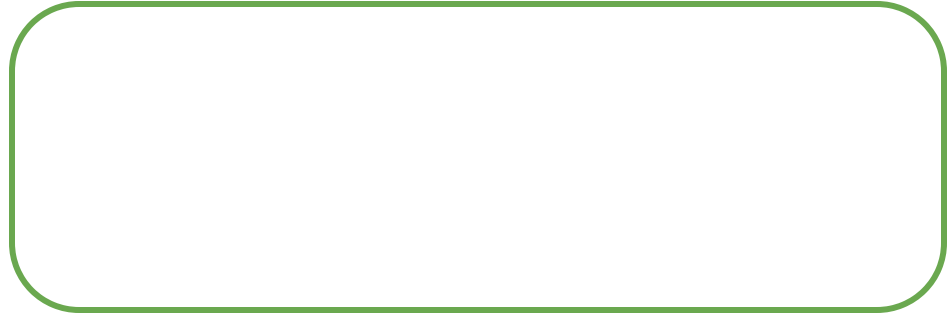
- **2 objectives** to minimize
- **2 freedom degree** / parameters (genome)
- **200 islands** (parallelism) of **5 minutes** each
- **20000 evolutions** performed by all the islands (200 island * 100 iteration)
- **1 hook** to save population

```
1  NSGA2Evolution(  
2    evaluation = model,  
3    genome = Seq(  
4      myParameter1 in (0.0, 1.0),  
5      myParameter2 in (-1.0,1.0)  
6    ),  
7    objectives = Seq(myObjective1,  
8      myObjective2),  
9    termination = 20000,  
10   parallelism = 200,  
11   distribution = Island(5 minutes),  
12   stochastic = Stochastic(seed = seed)  
13 ) hook (workDirectory /  
14   "results/example") on env
```

ZombieLand

Application to Zombieland

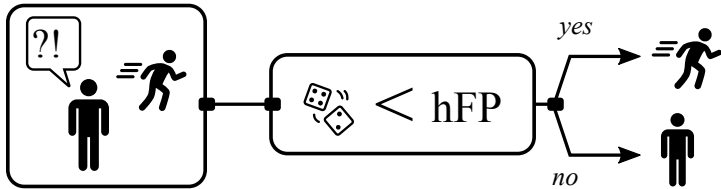
Trying to calibrate free parameters !



```
1 val humanFollowProbability = Val[Double]
2 val humanInformedRatio = Val[Double]
3 val humanInformProbability = Val[Double]
4
5 val result = zombieInvasion(
6   humanFollowProbability = humanFollowProbability,
7   humanInformedRatio = humanInformedRatio,
8   humanInformProbability = humanInformProbability,
9   zombies = 4,
10  humans = 250,
11  steps = 500,
12  random = rng)
```


Application to Zombieland

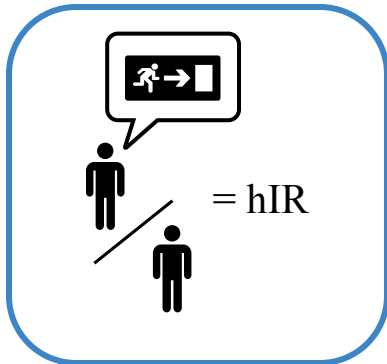
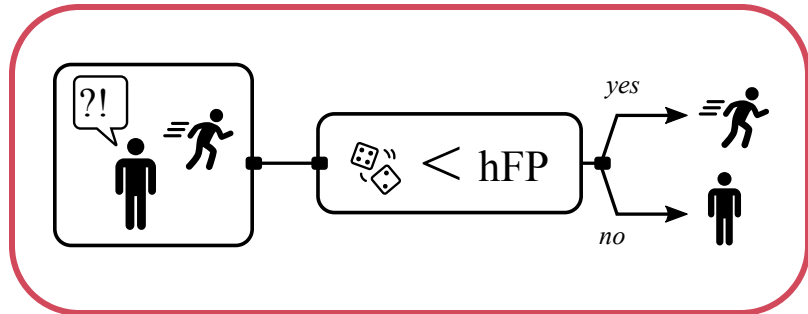
Trying to calibrate free parameters !



```
1 val humanFollowProbability = Val[Double]
2 val humanInformedRatio = Val[Double]
3 val humanInformProbability = Val[Double]
4
5 val result = zombieInvasion(
6   humanFollowProbability = humanFollowProbability,
7   humanInformedRatio = humanInformedRatio,
8   humanInformProbability = humanInformProbability,
9   zombies = 4,
10  humans = 250,
11  steps = 500,
12  random = rng)
```

Application to Zombieland

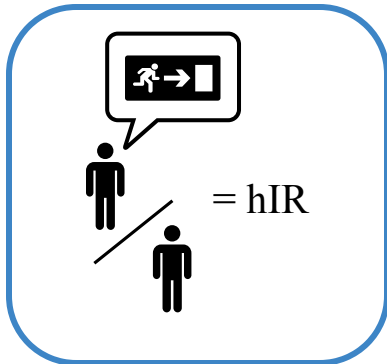
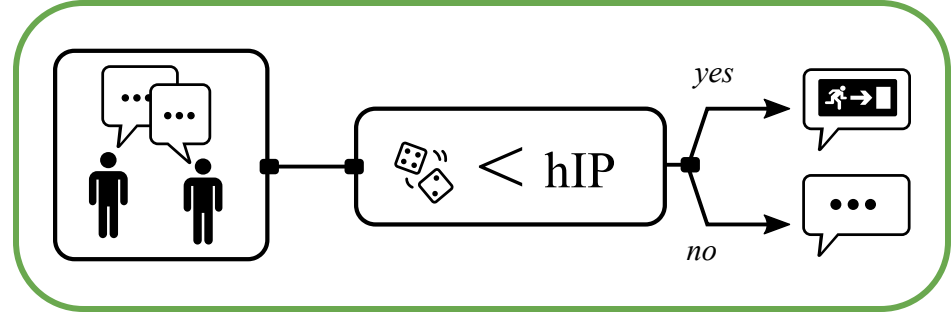
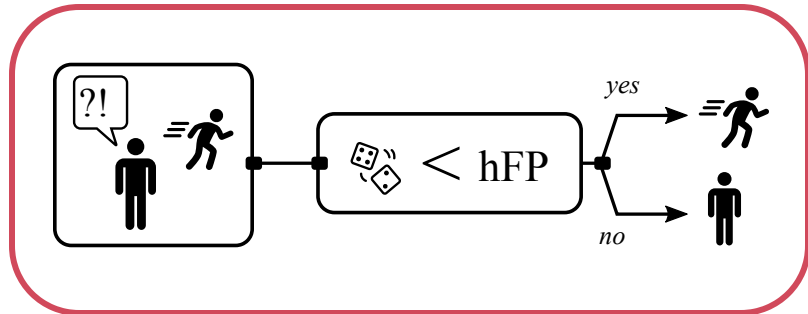
Trying to calibrate free parameters !



```
1 val humanFollowProbability = Val[Double]
2 val humanInformedRatio = Val[Double]
3 val humanInformProbability = Val[Double]
4
5 val result = zombieInvasion(
6   humanFollowProbability = humanFollowProbability,
7   humanInformedRatio = humanInformedRatio,
8   humanInformProbability = humanInformProbability,
9   zombies = 4,
10  humans = 250,
11  steps = 500,
12  random = rng)
```

Application to Zombieland

Trying to calibrate free parameters !

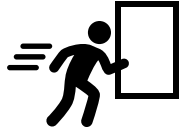


```
1 val humanFollowProbability = Val[Double]
2 val humanInformedRatio = Val[Double]
3 val humanInformProbability = Val[Double]
4
5 val result = zombieInvasion(
6   humanFollowProbability = humanFollowProbability,
7   humanInformedRatio = humanInformedRatio,
8   humanInformProbability = humanInformProbability,
9   zombies = 4,
10  humans = 250,
11  steps = 500,
12  random = rng)
```

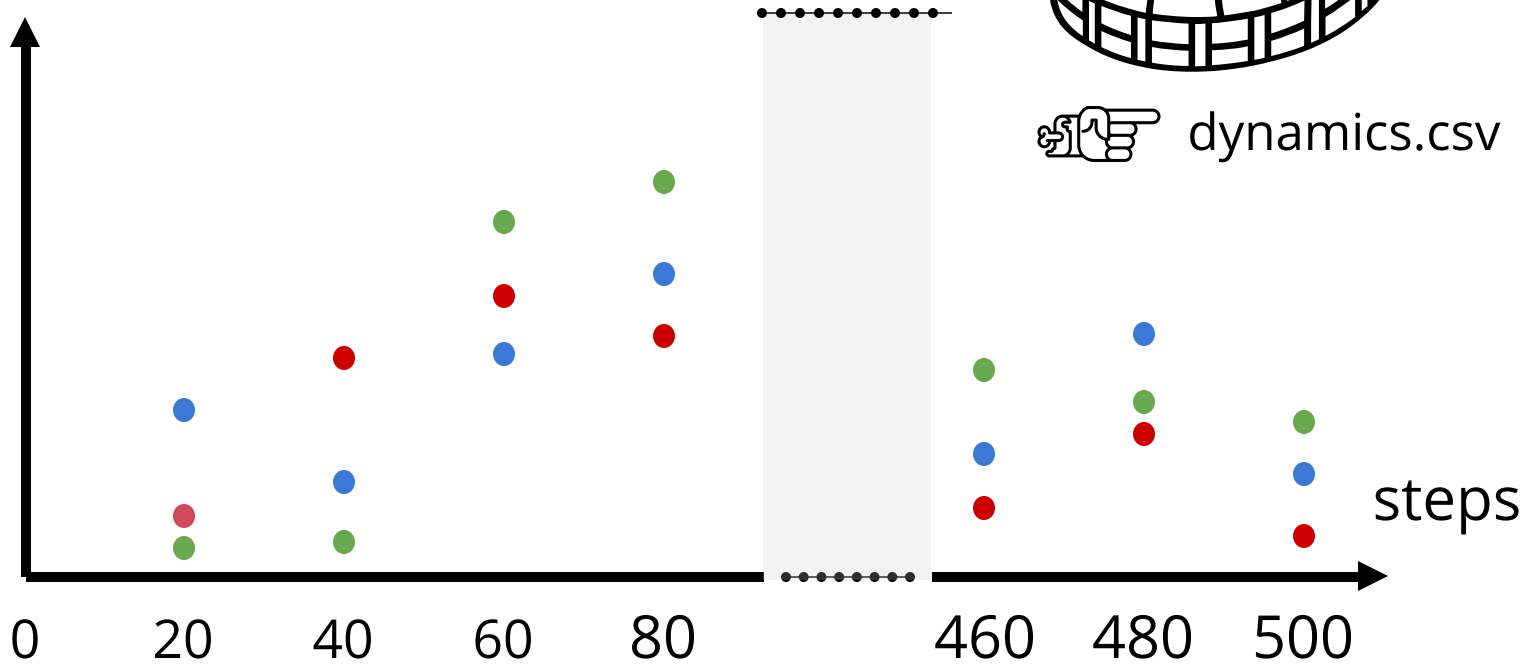
Application to Zombieland

Data Data Data !

evacuated



 dynamics.csv

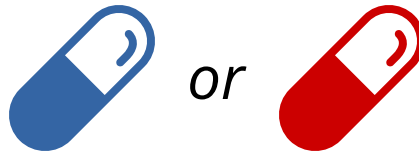


observed
evac.

2019 2020 2029 2030

Application to Zombieland

Road to Objective : strategy to compare data (real/sim.) ?



| step evac. | 20 | 40 | ... | 500 |
|---------------|----|----|-----|-----|
| 1 | | | | |
| 2 | | | | |
| ... | | | | |
| 100 | | | | |

median by
evac.

| |
|--|
| |
| |
| |
| |

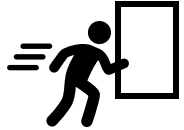
median by
step

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

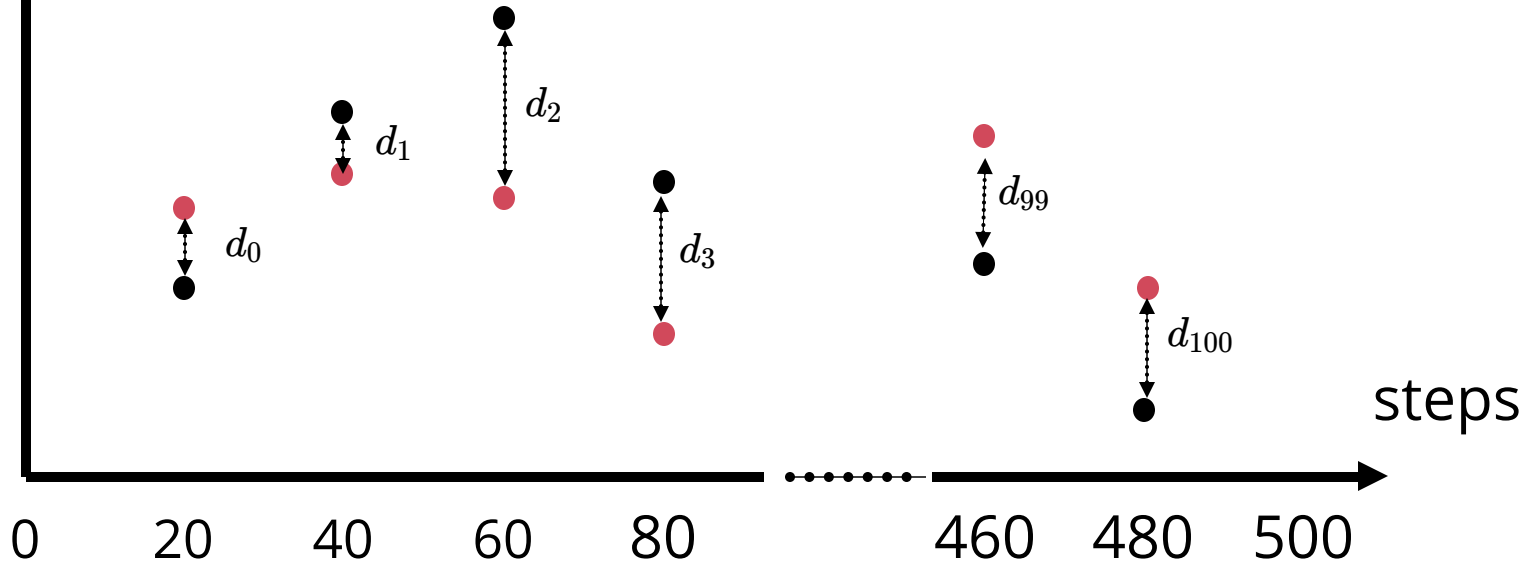
Application to Zombieland

Road to Objective : simulated vs real data ? 

evacuated



$$O_{calibrage} = \sum d_0..d_{100}$$

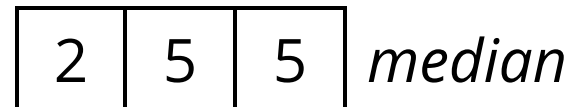
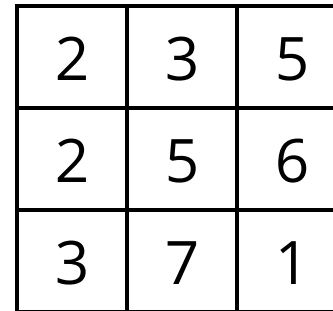
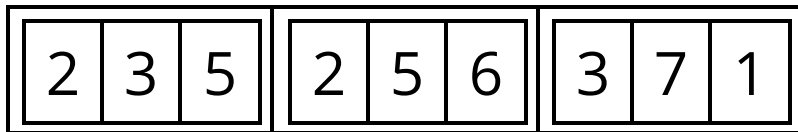


● real ● simulated → *search in metrics*

Application to Zombieland

OpenMolisation : function to aggregate simulated data

Vector[Array[Int]]



transpose exemple

```
1 val v: Vector[Array[Int]] =  
2   Vector(Array(2, 3, 5),  
3           Array(2, 5, 6),  
4           Array(3, 7, 1))  
5  
6 println(v.transpose)  
7 // return Vector(Vector(2, 2, 3),  
8                   Vector(3, 5, 7), Vector(5, 6, 1))  
9  
10 println(v.transpose.map(column =>  
11   column.sum))  
12 // return Vector(7, 15, 12)  
13  
14 println(v.transpose.map(column =>  
15   column.median))  
16 // return Vector(2, 5, 5)
```

function to define in wf

```
1 def distance(dataSim: Vector[Array[Int]]) = {  
2   realData = Array( ... ) // my median on real  
3   data  
4   absoluteDistance ( dataSim, ... ) // obj  
5 }
```



Application to Zombieland

OpenMolisation : the last bricks

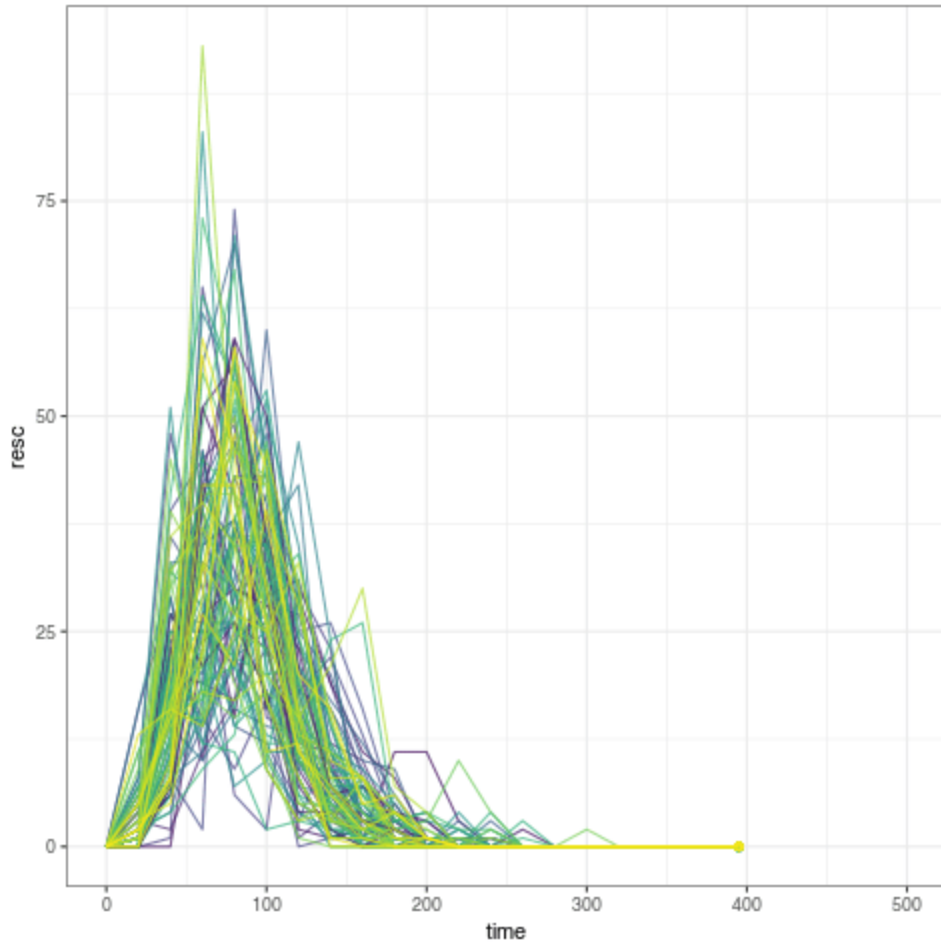
EA definition with calibrate objective

```
1 NSGA2Evolution(  
2   evaluation = model ,  
3   genome = Seq(  
4     humanInformedRatio in (0.0, 1.0),  
5     humanInformProbability in (0.0, 1.0),  
6     humanFollowProbability in (0.0, 1.0)  
7   ),  
8   objectives = Seq(rescuedDynamic aggregate distance),  
9   termination = 20000,  
10  parallelism = 200,  
11  distribution = Island(5 minutes),  
12  stochastic = Stochastic(seed = seed)  
13 ) hook (workDirectory / "results_calibration/distance",  
    frequency = 20) on env
```

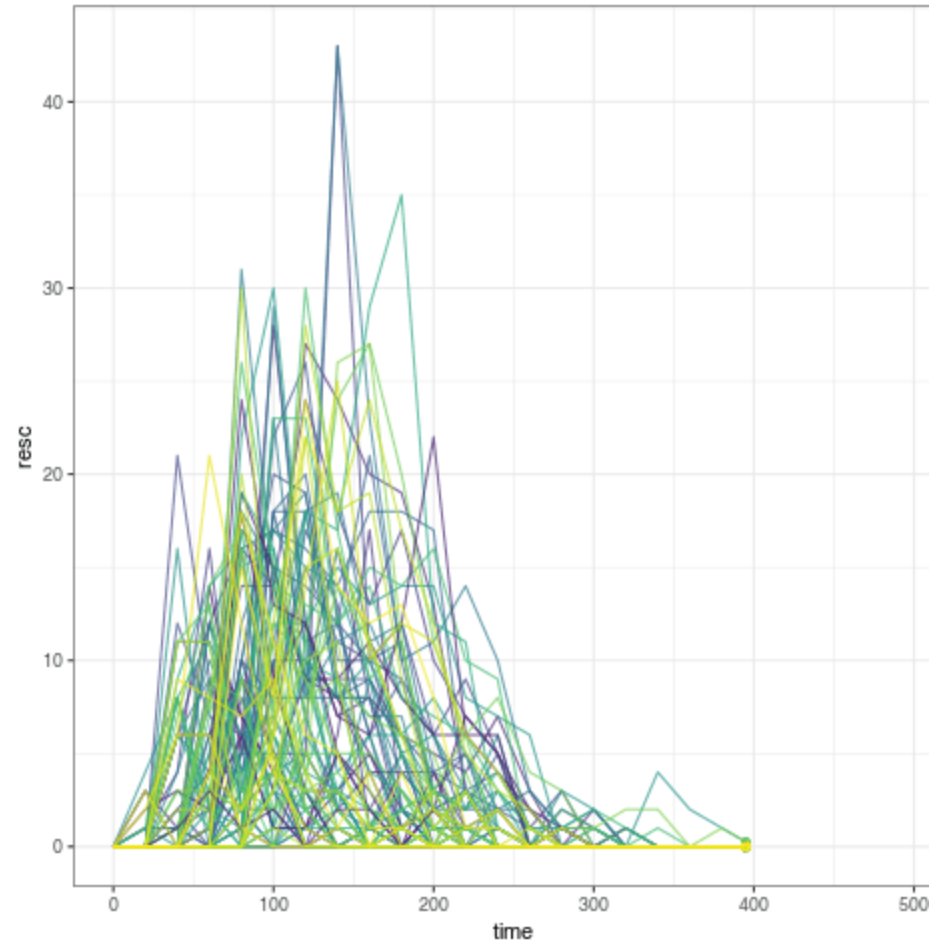

Application to Zombieland

Results

Real dynamics



Simulated dynamics



Application to Zombieland

Wait a sec ...

- optimize perform with sum of distance criterium
- doesn't capture the overall dynamic to it's full extent

- Open question :
 - introduce additional criterium ?
 - change the aggregate function (MSE ?)

Application to Zombieland

Public policies, try to anticipate by optimisation



Training

- + faster
- endurance

$P_{runFast}$



Learning

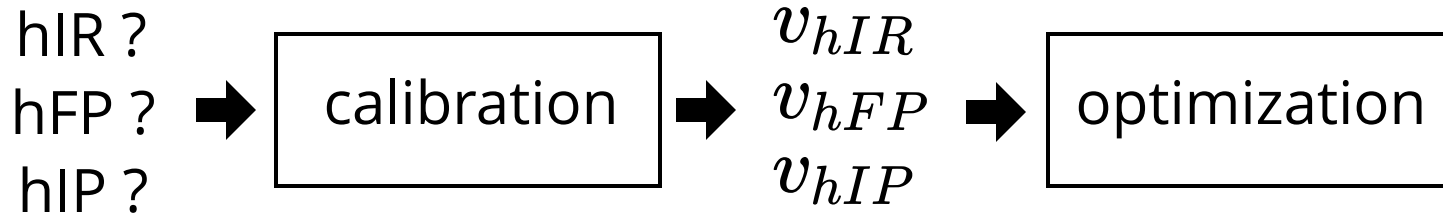
- + informed people
- communication skill

$P_{formInform}$



Application to Zombieland

Public policies, try to predict by optimisation



```
1 val result = zombieInvasion(  
2   humanFollowProbability = vhFP,  
3   humanInformedRatio = vhIR + (formInform * vhIR),  
4   humanInformProbability = vhIP - (formInform * vhIP),  
5   humanExhaustionProbability = physic.humanExhaustionProbability - (runFast *  
   physic.humanExhaustionProbability),  
6   humanRunSpeed = physic.humanRunSpeed + (runFast * physic.humanRunSpeed),  
7   zombies = 4,  
8   humans = 250,  
9   steps = 500,  
10  random = rng)
```

Optimal values from
previous calibration

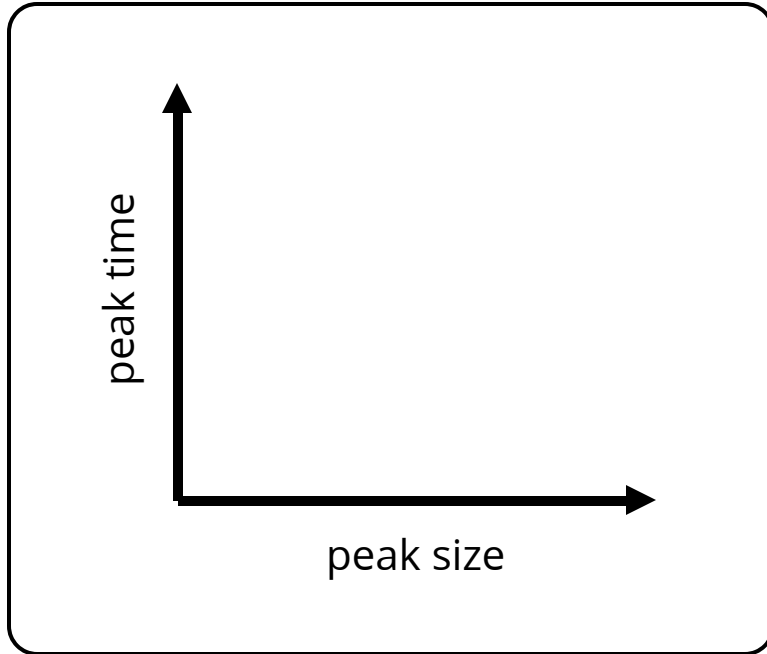
$v_{hFP} = 0.13234604715266718$
 $v_{hIR} = 0.1442621096357345$
 $v_{hIP} = 0.07780857466116062$



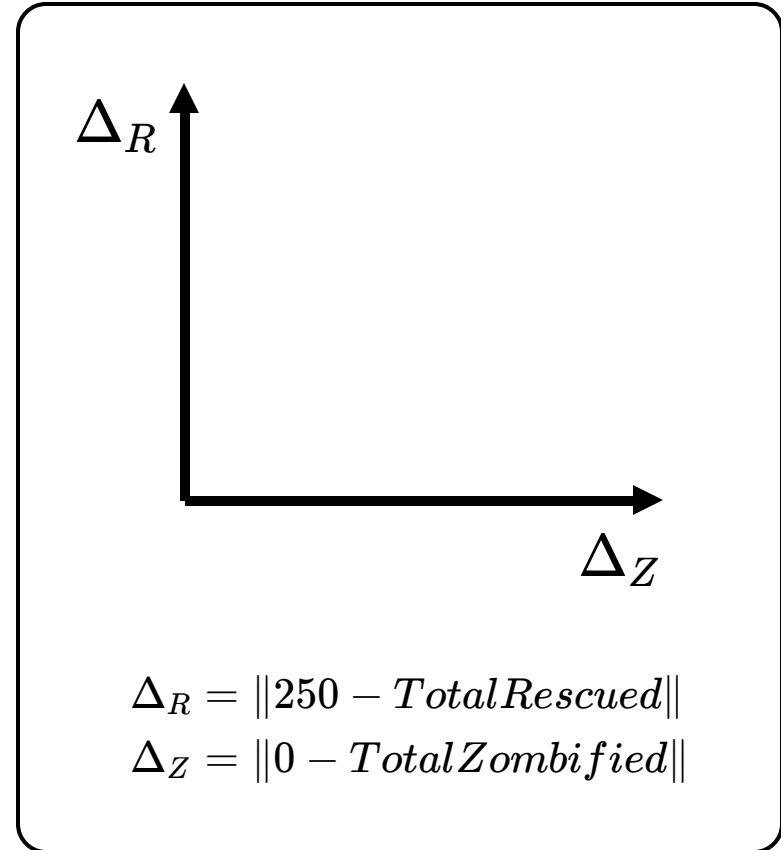
new mechanism for
public policy !

Application to Zombieland

Public policies, try to anticipate by optimisation



opti 1 : limit epidemic

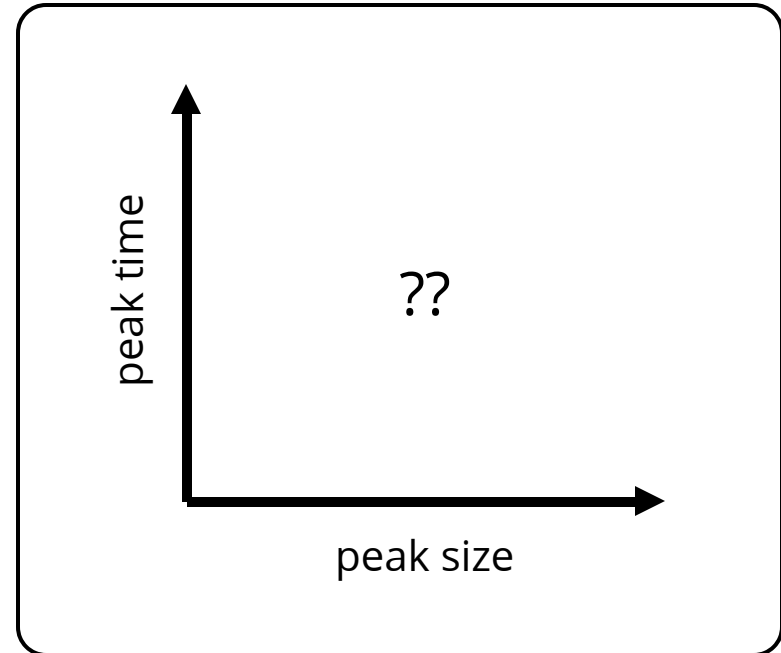


opti 2 : save lives

Application to Zombieland

OpenMolisation : limit epidemy

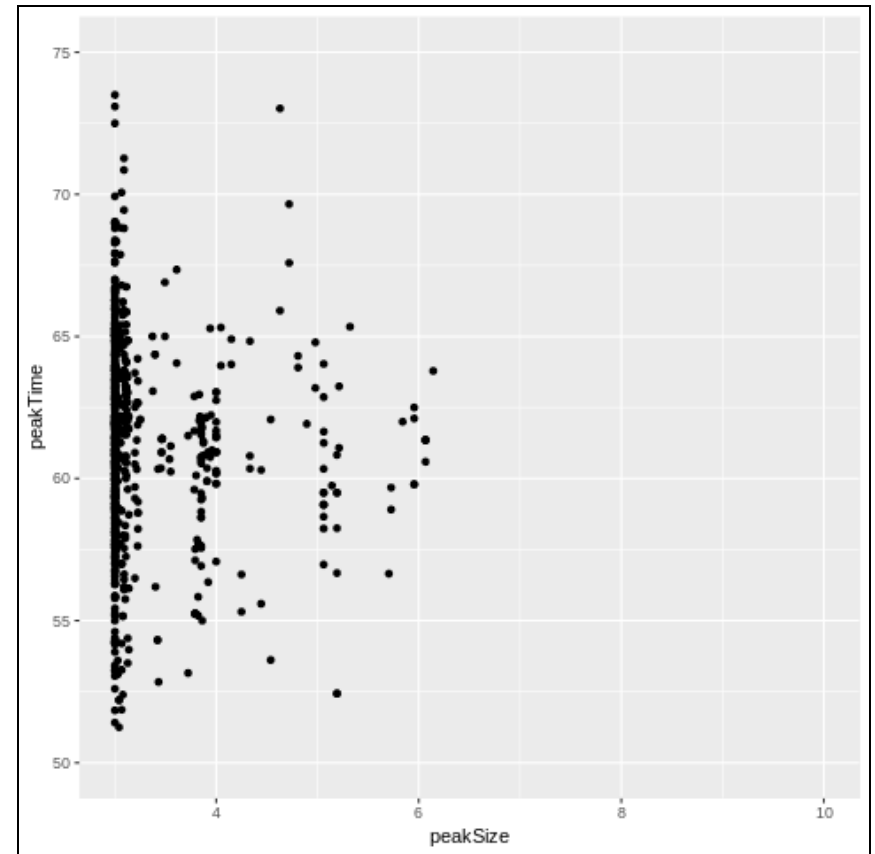
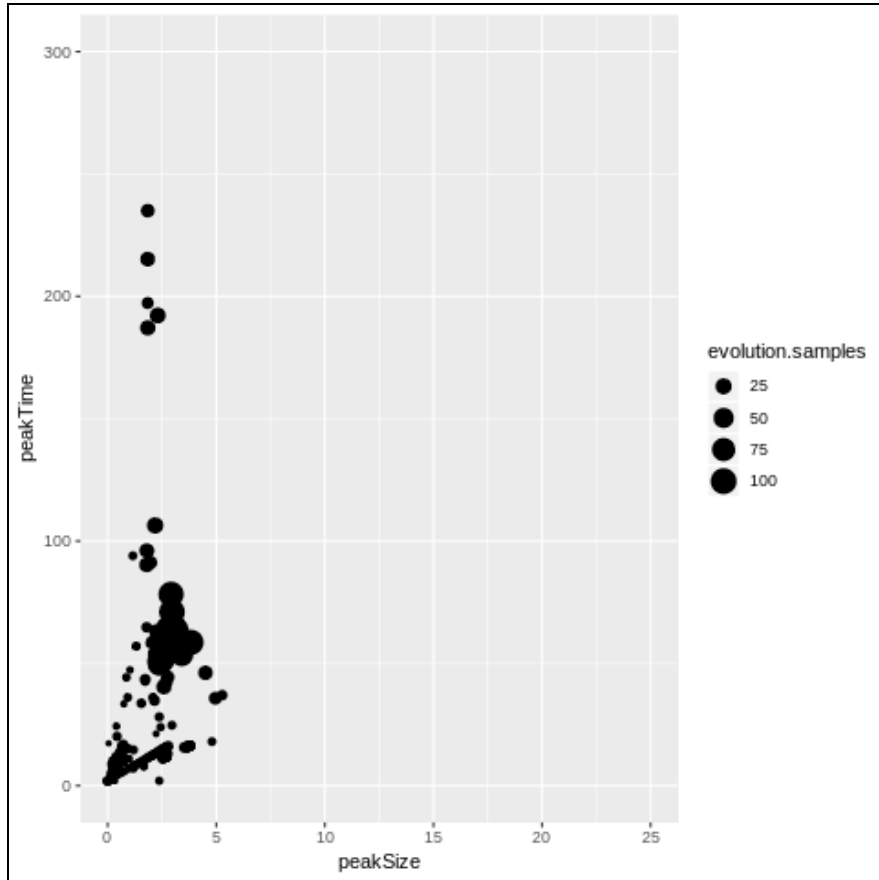
```
1 NSGA2Evolution(  
2   evaluation = model,  
3   genome = Seq(  
4     runFast in (-1.0,1.0),  
5     formInform in (-1.0,1.0)  
6   ),  
7   objectives = Seq(peakTime,  
8     peakSize),  
9   termination = 20000,  
10  parallelism = 200,  
11  distribution = Island(5 minutes),  
12  stochastic = Stochastic(seed =  
13    seed)  
14 ) hook (workDirectory /  
15   "results_opti/opti1") on env
```



limit epidemy

Application to Zombieland

OpenMolisation : limit epidemy

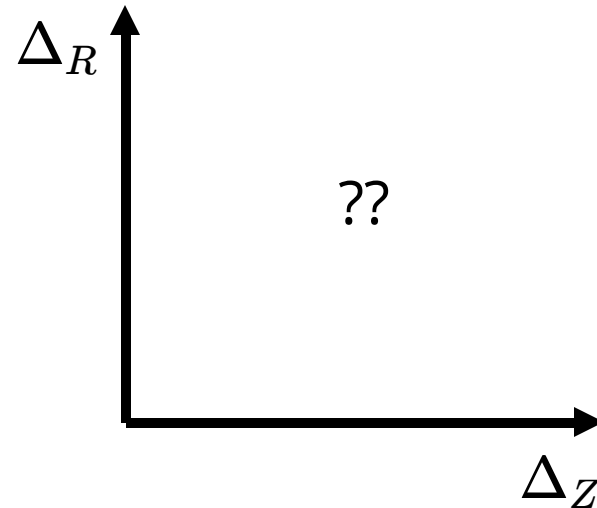


Zoom

Application to Zombieland

OpenMolisation : save lifes

```
1 NSGA2Evolution(  
2   evaluation = model --  
   DeltaTask(totalZombified -> 0,  
   totalRescued -> 250),  
3   genome = Seq(  
4     runFast in (-1.0,1.0),  
5     formInform in (-1.0,1.0)  
6   ),  
7   objectives = Seq(totalZombified,  
   totalRescued),  
8   termination = 20000,  
9   parallelism = 200,  
10  distribution = Island(5  
   minutes),  
11  stochastic = Stochastic(seed =  
   seed)  
12 ) hook (workDirectory /  
   "results_opti/opti2") on env
```



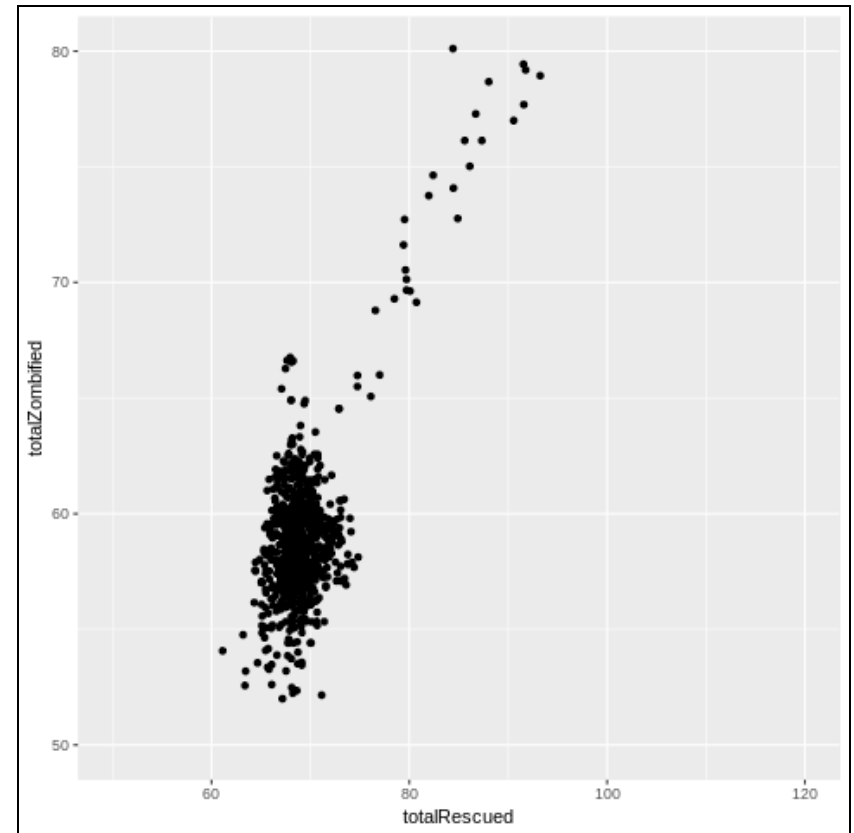
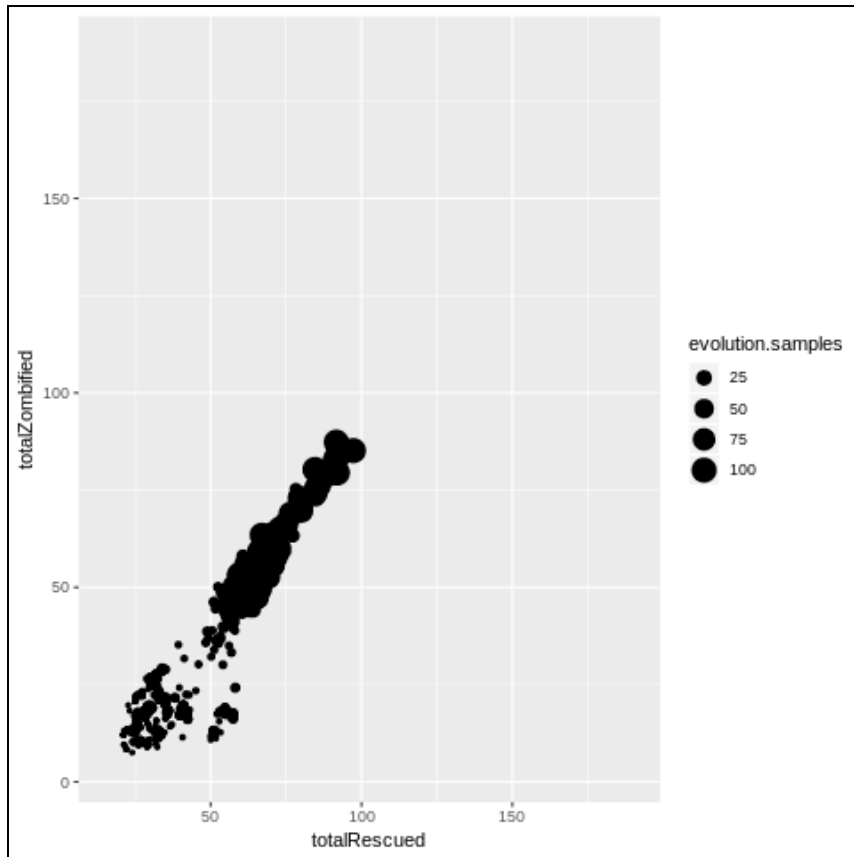
$$\Delta_R = \|250 - TotalRescued\|$$

$$\Delta_Z = \|0 - TotalZombified\|$$

save lifes

Application to Zombieland

OpenMolisation : save lifes



Zoom

Application to Zombieland

Interpretation

- No pareto front :
 - No need to choose a compromise
 - The criteria are compatibles
 - The criteria are either :
 - totally independent
 - in a "non constraining relationship"
- It still converge toward good solutions.

Going further

[Banos2016] Banos, A., Lang, C., & Marilleau, N. (2016). *Agent-based spatial simulation with NetLogo* Volume 2: Advanced Concepts. Elsevier. [url](#)

[Banos2016] Banos, Arnaud (2016) Modéliser c'est apprendre : Itinéraire d'un géographe, Edition Matériologique

[Chérel2015] Chérel G., Cottineau C., Reuillon R., 2015, « Beyond Corroboration : Strengthening Model Validation by Looking for Unexpected Patterns. », PLoS ONE 10(9), e0138212. [doi:10.1371/journal.pone.0138212](https://doi.org/10.1371/journal.pone.0138212)

[Cottineau2015] Cottineau C., Reuillon R., Chapron P., Rey-Coyrehourcq S., Pumain D., 2015, "A Modular Modelling Framework for Hypotheses Testing in the Simulation of Urbanisation.", *Systems*, 3, Numéro Spécial "Agent-Based Modelling of City Systems", 348-377. DOI : [10.3390/systems3040348](https://doi.org/10.3390/systems3040348)

[Cottineau2015] Cottineau C., Chapron P., Reuillon R., 2015, "Growing models from the bottom up. An evaluation-based incremental modelling method (EBIMM) applied to the simulation of systems of cities", *Journal of Artificial Societies and Social Simulation (JASSS)*, Vol. 18, No. 4, 9. DOI : [10.18564/jasss.2828](https://doi.org/10.18564/jasss.2828).

[Cottineau2016] Cottineau, Clémentine, Rey-Coyrehourcq Sébastien (2016) "Back to the future of multi modelling" , Conférence RGS , [slides](#)

[Delay2015] Delay E., « Réflexions géographiques sur l'usage des systèmes multi-agents dans la compréhension des processus d'évolution des territoires viticoles de fortes pentes : Le cas de la Côte Vermeille et du Val di Cembra », Thèse de doctorat, Université de Limoges, Limoges, 2015. [HAL-SHA](#)

[Rey2015] REY-COYREHOURCQ, Sébastien (2015, October 13) "Une plateforme intégrée pour la construction et l'évaluation de modèles de simulation en géographie" Thèse Paris 1. <https://zenodo.org/record/50212>

[Reuillon2015] Reuillon, R., Schmitt, C., De Aldama, R., & Mouret, J.-B. (2015). A New Method to Evaluate Simulation Models: The Calibration Profile (CP) Algorithm. *Journal of Artificial Societies and Social Simulation*, 18(1), 12. Retrieved from <http://jasss.soc.surrey.ac.uk/18/1/12.html>

[Reuillon2013] Reuillon, R., Leclaire, M., & Rey-Coyrehourcq, S. (2013). OpenMOLE, a workflow engine specifically tailored for the distributed exploration of simulation models. *Future Generation Computer Systems*, 29(8), 1981–1990. <https://doi.org/http://dx.doi.org/10.1016/j.future.2013.05.003>

[Schmitt2015] Schmitt, C., Rey, S., Reuillon, R., & Pumain, D. (2015). Half a billion simulations: Evolutionary algorithms and distributed computing for calibrating the SimpopLocal geographical model. *Environment and Planning B*, 42(2), 300–315. [url](#)

[Thiele2014] Thiele, Jan C., Kurth, Winfried and Grimm, Volker (2014) 'Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models: A Cookbook Using NetLogo and 'R'' *Journal of Artificial Societies and Social Simulation* 17 (3) 11 <<http://jasss.soc.surrey.ac.uk/17/3/11.html>>. doi: 10.18564/jasss.2503

++ David O'Sullivan !! ++

++ Volker GRIMM !! ++