

Metaheuristics: glorious past, controversial present and bright future?

Marc Sevaux

President of EURO

Université de Bretagne-Sud
Lab-STICC – Lorient – FRANCE

September 2022

What this presentation is about:

- giving a **high-level** understanding of some metaheuristics
- showing the different **eras** of metaheuristics
- and **where we should not go**

What this presentation is not about:

- **deep-diving** into a particular approach
- **compare** the metaheuristics and **tell you** which one is the best
- give a **chronological** presentation of metaheuristics

Contents

- 1 A short introduction
- 2 The golden age of metaheuristics
- 3 The nature-inspired metaheuristics
- 4 The crossroad of metaheuristics and machine learning
- 5 Conclusions

Contents

- 1 A short introduction
 - Definitions
 - Greedy heuristic
 - Greedy randomised search procedure (GRASP)
- 2 The golden age of metaheuristics
- 3 The nature-inspired metaheuristics
- 4 The crossroad of metaheuristics and machine learning
- 5 Conclusions

Introduction to metaheuristics

Etymology

Heuristic comes from the Greek verb *heuriskein*, “to search” or “to discover”. Eureka (from heureka), “I’ve found it !”



Heuristic

A **heuristic** is an **approach** that **quickly** finds a **feasible solution**, **not necessarily optimal**, for a hard optimization problem. Generally a heuristic is **problem-specific**.

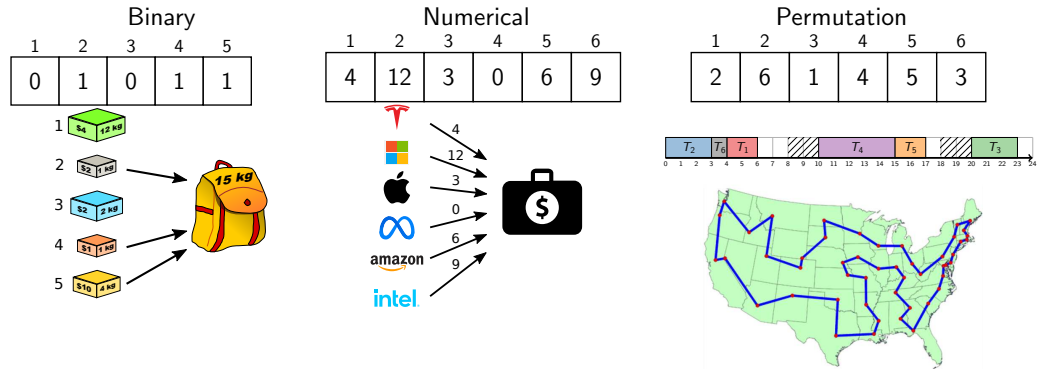
A heuristic is often the first step of a metaheuristic. . .

Metaheuristic

A **metaheuristic** is a **problem-independent high-level strategy** designed to guide a search in order to solve a **larger spectrum of problems**

Encoding

A **real problem** always needs to be **converted** into a **exploitable mathematical formulation**. The **encoding** is a **computer-based representation** of a **solution** of a given problem (array, matrix, integers, etc.)



Greedy heuristic

A greedy heuristic starts from an **empty solution** and **iteratively adds solution components** one by one and **locally optimum** until the solution is complete

Procedure Greedy

```

s ← ∅
while s incomplete do
  x ← arg min_{y ∈ X} f(s ∪ {y})
  s ← s ∪ {x}

```

Output s

Advantages : **simple** to implement and **fast**

Disadvantages : **never question** the previous choices
poor performance and always reports the same solution for the same data

GRASP: greedy randomized search procedure

To avoid repetition, **add randomness**



GRASP [Feo, Resende, 1995] is **identical** to the **greedy heuristic** **except** it introduces a **chance of not always selecting the locally optimal** solution component

- create a RCL and choose one element at random
- use a probability to skip the best ones recursively
- . . .

Advantages : **better** than **greedy** if **executed multiple times**

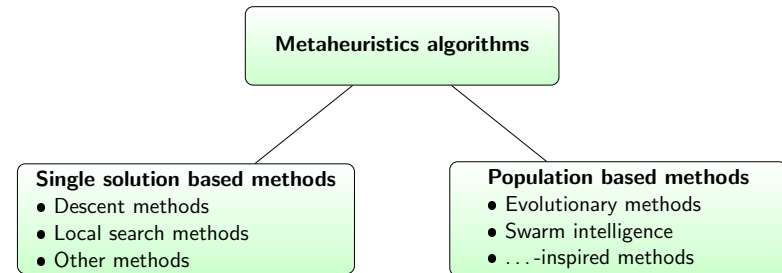
Disadvantages : still **weak performance** in general

Contents

- 1 A short introduction
- 2 The golden age of metaheuristics
 - History and classification
 - Descent methods
 - Escaping local optima
 - Population-based methods
 - Where to go?
- 3 The nature-inspired metaheuristics
- 4 The crossroad of metaheuristics and machine learning
- 5 Conclusions

The golden age

Metaheuristics are known as far as 1961 with **Pattern Search** from [Hooke and Jeeves, 1961] or better known with **Evolutionary Algorithms/Strategies** (EA-ES) since [Schwefel, 1965]

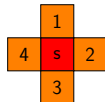


A metaheuristic should always balance **exploration** and **exploitation** also named **diversification** and **intensification**

Illustrative example

A **solution** s is represented by a **position** (x,y) in the grid to the right. Search for the minimum value

A **movement** consists in moving to an **adjacent** position (vertically or horizontally) in the **order** given below



We start from an **initial position** chosen at random and then we use the chosen **movement** in order to **explore** other solutions

	1	2	3	4	5	6	7	8	9	10
10	102	98	96	94	92	91	78	82	103	107
9	99	100	98	90	75	104	99	84	152	99
8	98	95	94	95	73	64	98	86	88	93
7	83	90	80	82	84	84	71	88	98	92
6	99	81	82	78	83	78	73	102	99	90
5	100	102	82	77	76	75	95	96	90	89
4	91	84	103	94	96	82	94	93	99	99
3	98	90	91	92	95	98	92	96	94	101
2	100	96	93	96	96	94	93	95	97	100
1	99	88	94	102	98	100	91	90	96	98

The simple descent

No better solution : local optimum

	1	2	3	4	5	6	7	8	9	10
10										
9										
8										
7										
6										
5		102								
4	91	84	103	94	96					
3		90	91	92	95	98				
2			93	96	96					
1					98					

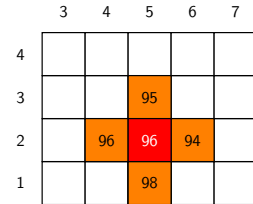
Deepest descent

It might be the case that the **first choice** of a neighboring solution is **not the best** one

During the second step of the previous execution we could have gone to a solution with 94 instead on 95

Exploring all neighbors could allow to find better solutions faster

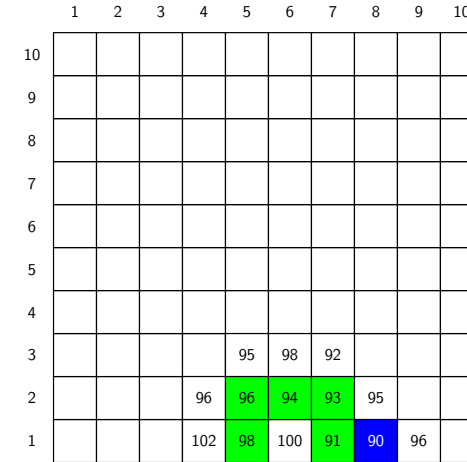
→ **deepest descent**



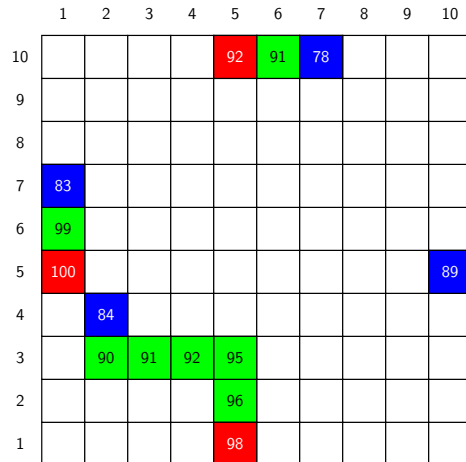
There is a cost to pay: **exploring** all neighbors of the current solution

Deepest descent

No better solution : local optimum



Multistart descent






Escaping local optima


At the end, we are always in a local optimum, because of

- the choice of the **encoding** and the **objective function**
- the **movement** for generating neighbors
- **the basic principle of descent methods**

To escape, we can

- add randomness  during the search
- accept degrading solutions temporary 
- change the neighborhood 

Simulated annealing

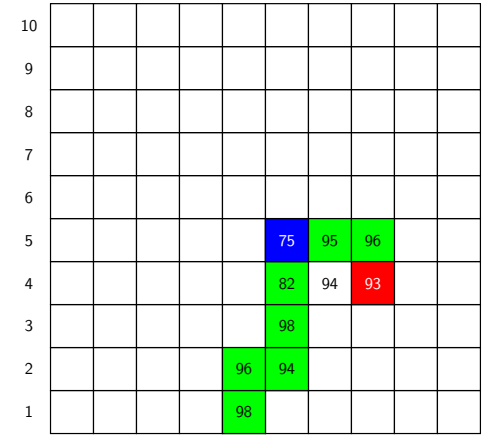
Proposed by [Kirkpatrick et al., 1983], **simulated annealing** adds **randomness**  during the exploration

Main steps:

- 1 Start with an initial solution s
- 2 Choose a random neighbor $s' \in N(s)$
- 3 Accept this neighbor if it is better than s or with probability $e^{-\Delta/T}$
- 4 Go back to step 2 until a stopping criterion is met

Simulated annealing

Stopping condition → retrieve best explored solution



Tabu search

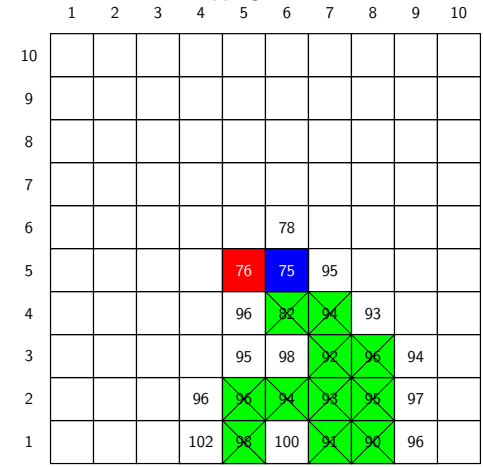
Proposed by [Glover, 1986], **tabu search** accepts degrading solutions during the search. It can be seen as a **deepest descent method** with **memory**

Main steps:

- 1 Start with an initial solution s
- 2 Choose the best neighbor $s' \in N(s)$, regardless of its objective function
- 3 Accept the neighbor if its not in the tabu list
- 4 Go back to step 2 until a stopping criterion is met

Tabu search

Stopping condition

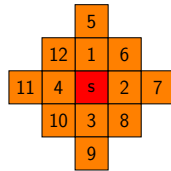
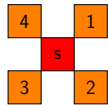
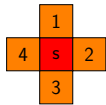


Variable Neighborhood Search in practice

Proposed by [Mladenović and Hansen, 1997], VNS change the neighborhood to avoid being blocked

Variable Neighborhood Search uses a **descent method** until reaching a **local optimum**, at which point it **switches the neighborhood** in order to progress further

- 1: Cross movement 2: Star movement 3: Two times the cross movement



Population-based approaches

Unlike the single-solution approaches, **population-based approaches** use a **set of solutions** to **explore** the **search space**.

This set of solutions is called a **population**

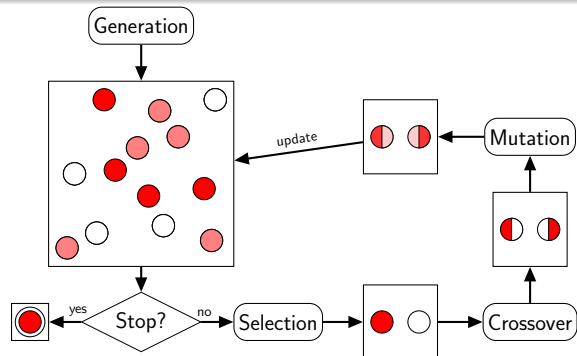
We can cite:

- Genetic algorithm (GA) [Holland, 1975, Goldberg, 1989]
- Memetic algorithm (MA) [Moscato, 1989]
- MA with population management (MA|PM) [Sörensen and Sevaux, 2006]
- Ant Colony Optimization (ACO) [Colomni et al., 1992]
- Scatter search (SS) [Glover, 1998, Glover et al., 2000]
- ...

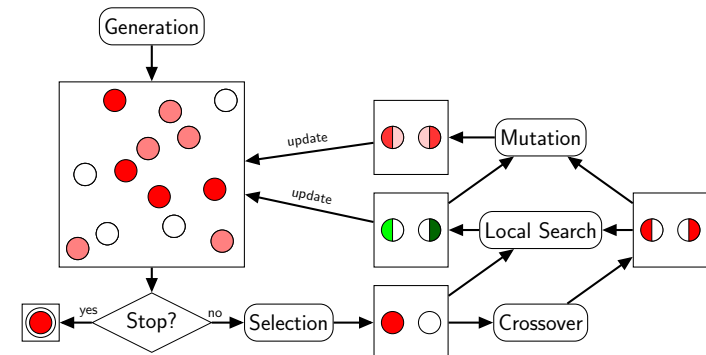
The concept of a genetic algorithm

Concept

The Darwinian principle of survival of the fittest



The concept of a memetic algorithm



Other methods

Many other population-based methods exist:

- Memetic algorithms with population management – MA|PM (MA + keep a small population of improved individuals)
- Ant colony optimization (multi-agent system based method)
- Scatter search (combine exhaustively all individuals and keep the best ones)
- ...

Any recommendations?

I said: “I will not tell you which metaheuristics is the best”

- 1 think about your problem and how you can model it → single solution/population
- 2 start with something modest and easy → descent methods
- 3 improve step by step → refine your neighborhood(s)
- 4 go further → from descent to tabu search, SA or VND
- 5 add randomness → where is your GRASP?
- 6 repeat → multistart

My best advice: **Multistart GRASP + VND**

Contents

- 1 A short introduction
- 2 The golden age of metaheuristics
- 3 **The nature-inspired metaheuristics**
 - Why, where
 - Creative innovation
 - Fighting
- 4 The crossroad of metaheuristics and machine learning
- 5 Conclusions

The nature-inspired metaheuristics

A new tendency

Following the publication of Ant Colony Optimization (1992), many authors have decided to develop new **nature inspired** metaheuristics

Just a few:

- 1993 Great deluge algorithm
- 1995 Bachelors
- 2000 Antibodies
- 2001 Bees; Harmony search; Sheep flocks
- 2002 Bacteria; Gene expression
- 2003 Fish swarm; Frog leaping; Social behavior
- 2005 Big bang; Dendritic cells; Nuclear collision; Wasps

...

Why do we have all these methods?

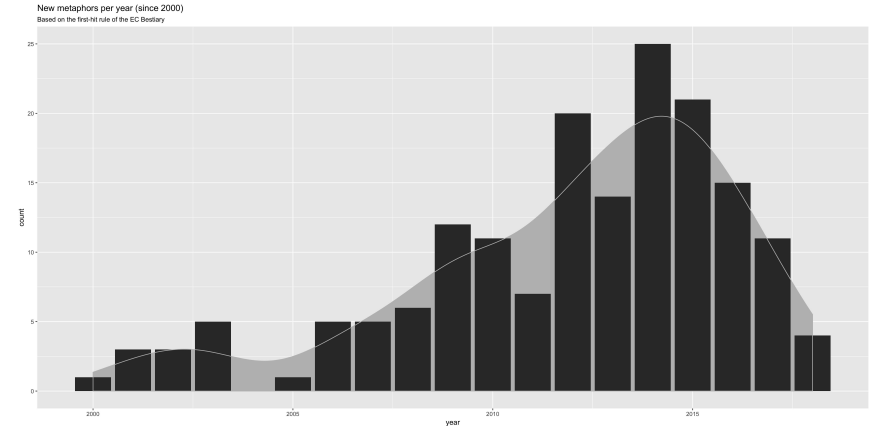
Yes, why?

- you want a method that fits your needs
- that solve your problem efficiently
- that is and will stay the best to solve your problem
- you want to publish it
- and it's fun :-)

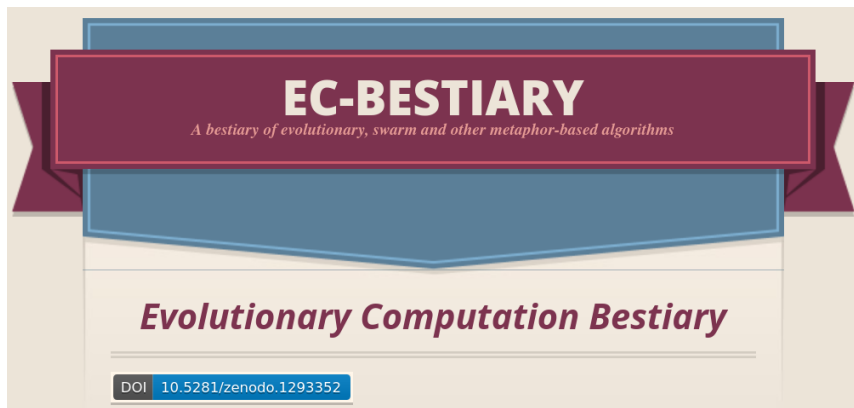
Like many researchers in the field, we have started with the classical methods and we wanted to improve them. But suddenly things went crazy...

Ruben Ruiz in 2018 called it the **bio-inspired tsunami**...

Too many of them



Where to find them?



<http://fcampelo.github.io/EC-Bestiary/>

Wanna play?



<http://fcampelo.github.io/EC-Bestiary/>

A few examples I

Red fox (2021)

Optimization paradigm based on mathematical model of red fox life and hunting behavior. Efficient mechanisms of movement, hunting and developing population

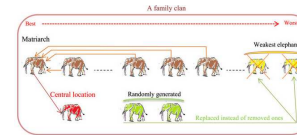


We can see that results of our method are related to the results of well-known heuristic algorithms like PSO, GA, AACA, and the latest nature-inspired algorithms like DA, FPA, MFO

A few examples II

Elephant herding Optimization (2015)

A new kind of swarm-based metaheuristic search method. EHO method is inspired by the herding behavior of elephant group. In nature, the elephants belonging to different clans live together under the leadership of a matriarch, and the male elephants will leave their family group when they grow up. These two behaviors give: clan updating operator and separating operator

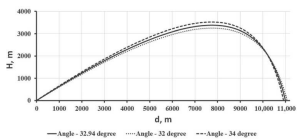


By comparing with BBO, DE and GA, EHO is benchmarked by fifteen test cases. EHO can find much better solutions on most benchmark problems than three other algorithms

A few examples III

Algorithm of innovative gunner (2019)

Inspired by the choice of artillery parameters that sends a shot precisely to a target. The applied correction of the cannon's setting parameters **differs from that defined by classic artillery theory**. Characterized by high efficiency and speed in solving problems having nothing to do with ballistics



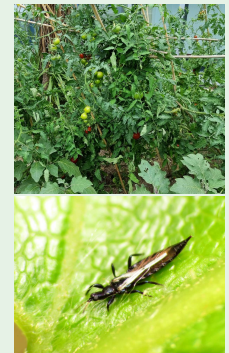
The obtained results prove its high efficiency and competitiveness against known swarm intelligence methods, such as PSO, GA, CS, GSA, GWO, ABC and BBO

Let's create a new method

The Tomato and Thrip Optimizer Algorithm – TTOA (2022)

Last week, during a walk close to the university, I've seen these tomato plants and small insects around, and then **EUREKA**, I had the idea to create a new nature-inspired metaheuristic...

- tomatoes are solutions, the redder the better
- thrips are arriving with wind gust (initial solutions)
- thrips are moving from one tomato to another
- thrips finding better fruits "calls" others
- wind gusts make thrips move to different positions
- ...



I am a serious researcher

- I draw fancy figures with tomatoes, arrows, moving thrips to explain the method
- I put a bit of maths (x, y, z, α, β) to make it serious
- I select carefully one or two completely new problems (engineering problems are fashionable today)
- I select a bunch of other methods from the EC-bestiary
- I run experiments (and ensure that I get the best results among other methods)
- I cite a lot of papers with nature-inspired methods (especially those from the journal I want to submit; no need to have them related to my problems)
- I submit the paper in the field of the application problems (not much competition and no/few knowledge of potential reviewers)

And finally

Consequences

- 1 I have a new publication (maybe even in a journal with high impact factor)
- 2 I have increased the citation of others (and thankfully they will increase mine later on)
- 3 I have contributed to the bio-inspired tsunami. . .

Is it good research?

Going further → write a survey

ISSN 2347 - 3983



Volume 8, No. 2, February 2020

International Journal of Emerging Trends in Engineering Research

Available Online at <http://www.warse.org/IJETER/static/pdf/file/ijeter29822020.pdf>
<https://doi.org/10.30534/ijeter/2020/29822020>

An Exploratory Retrospective Assessment on the Usage of Bio-Inspired Computing Algorithms for Optimization

Omdev Dahiya¹, Kamna Solanki², Sandeep Dalal³, Amita Dhankhar⁴

^{1,2,4}University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak-124001, India

³Department of computer science and applications, Maharshi Dayanand University, Rohtak-124001, India

¹Omdevdahiya21792@gmail.com, ²Kamna.mdurohtak@gmail.com

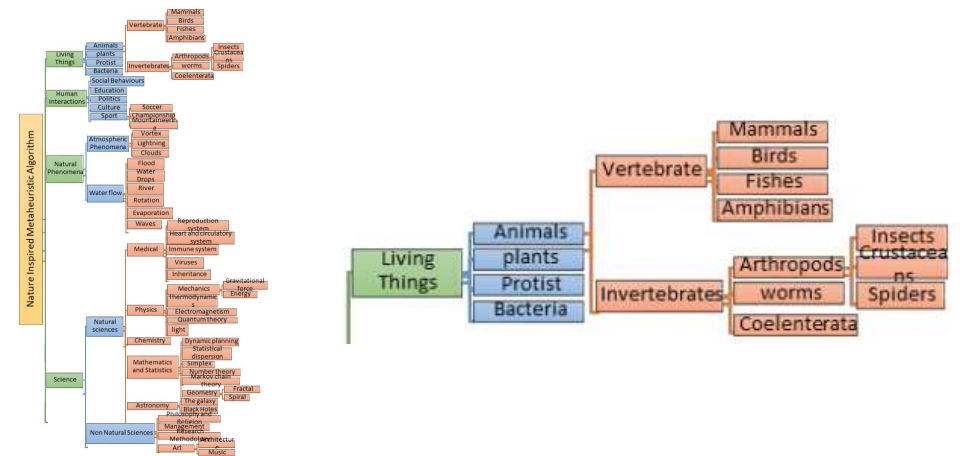
³Sandeepdalal.80@gmail.com, ⁴Amita.infotech@gmail.com

ABSTRACT

In the era of digitalization, every task is performed with the help of software-dependent applications. Therefore, the

worldwide. There is an exchange of information, technology, and exchange of goods and services. The advancements in digital technology boost this. Now we can get anything at our doorstep just by making a few clicks. The researchers have developed various software-based applications. With so much

And beyond → make a new classifications



But, have you heard of the

No Free Lunch theorem

A general-purpose universal optimization strategy is impossible, and the only way one strategy can outperform another is if it is specialized to the structure of the specific problem under consideration

In other words, if I spend enough time on tuning my algorithm, I will beat yours. . .

What should we do to avoid this?

We can fight the bio-inspired tsunami:

- use classical templates before searching for novelty
- do not invent a new nature inspired metaheuristics
- **do not let your colleagues do that too**
- if you are requested as a reviewer, reject all those papers
- **deconstruct the methods**
- ask for source code to make fair comparisons

Question yourself

Swarm Intelligence (2022) 16:1–6
<https://doi.org/10.1007/s11721-021-00202-9>



Metaphor-based metaheuristics, a call for action: the elephant in the room

Claus Aranha¹ · Christian L. Camacho Villalón² · Felipe Campelo³ · Marco Dorigo² · Rubén Ruiz⁴ · Marc Sevaux⁵ · Kenneth Sörensen⁶ · Thomas Stützle²

Received: 29 July 2021 / Accepted: 28 August 2021 / Published online: 30 November 2021
 © The Author(s) 2021

Taking inspiration from natural behaviors to devise new optimization algorithms has played an important role in the history of the field of metaheuristics (Sörensen et al. 2017)

Contents

- 1 A short introduction
- 2 The golden age of metaheuristics
- 3 The nature-inspired metaheuristics
- 4 The crossroad of metaheuristics and machine learning
 - 3 directions
 - Some work
- 5 Conclusions

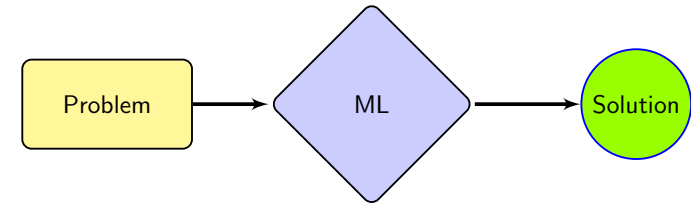
Metaheuristics and Machine Learning, the new friends. . .

Machine learning and metaheuristics live separately since decades, but it seems obvious today that they can cooperate

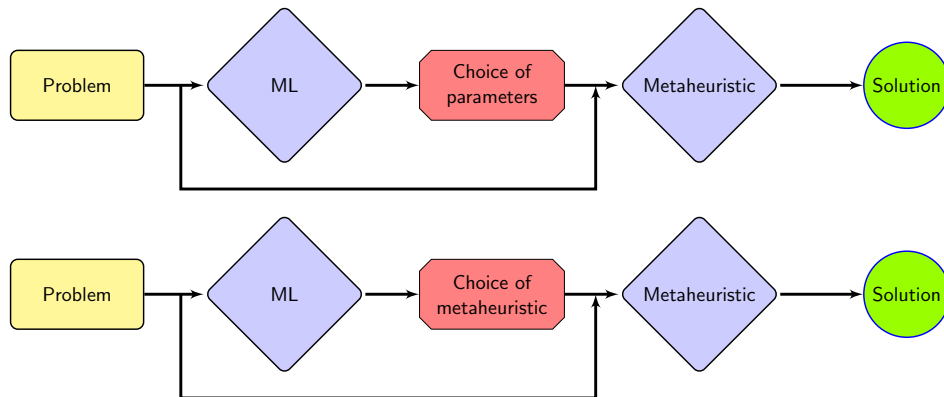
Many attempts have been made in various directions, but we can focus on the seminal paper from [Bengio et al., 2021] which presents 3 directions:

- learn to solve
- learn to configure
- learn to search

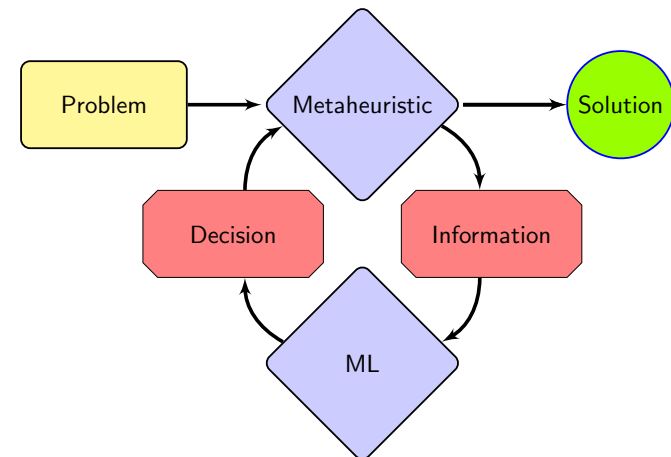
Learn to solve



Learn to configure



Learn to search



Recent work

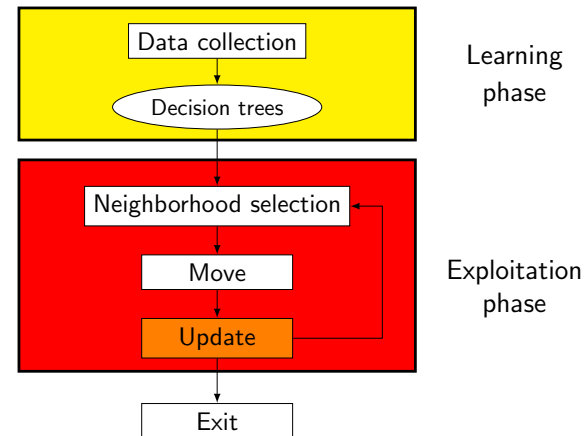
Experiments on MH/ML combination

- Flavien LUCAS' PhD thesis (Nov. 2020)
- urban logistic problems (VRP in cities with traffic)
- combination of ML and metaheuristics
- learning phase before the metaheuristics
- continue learning during execution

Results

- adaptive solver
- competitive, but **not enough** for winning the race
- learn some on instances and a lot on solutions

Feature-Guided Multiple Neighborhood Search – FGMNS



What's next?

- Very promising results
- Project "MAMUT 2023"
500k€ on 4 years
- 2 PhDs + 1 Post-doc to be hired

Contents

- 1 A short introduction
- 2 The golden age of metaheuristics
- 3 The nature-inspired metaheuristics
- 4 The crossroad of metaheuristics and machine learning
- 5 **Conclusions**

Conclusions

What we should do...

- classical frameworks of metaheuristics are still valid
- remember the **No Free Lunch Theorem**
- make fair comparisons / ready to share your code?
- explore the potential of machine learning

What you should not do...

- invent a new bio-inspired metaheuristic
- search for easy publications
- make this bio-inspire tsunami grow more