

Stepping Stones in Evolutionary Search

Characterisation and Challenges

Xavier F. C. Sánchez Díaz

Department of Computer Science
Faculty of Information Technology and Electrical Engineering
Norwegian University of Science and Technology



June 11, 2026

Slides available

<https://s.ntnu.no/trial-lecture>



<https://s.ntnu.no/trial-lecture>

Outline

- 1 Introduction: Search and Evolutionary Computation
- 2 Why are EAs *good* at search?
- 3 Problem Characterisation
- 4 Challenges in Search

Section 1

Introduction: Search and Evolutionary Computation

Search 101

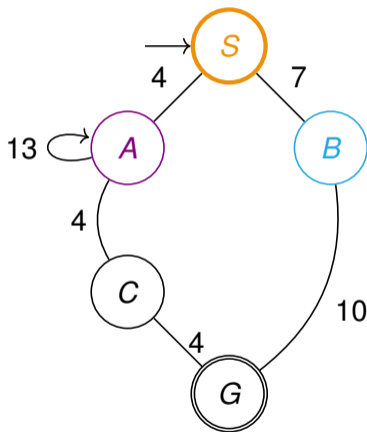
Introduction

Find a sequence of actions to go from the start state S to the goal state G .

- ▶ Actions have associated costs
- ▶ A heuristic function helps to evaluate which state to go to next

Sometimes we are only interested in finding the **goal state**.

This is why we do **local search**!



Everything is Search

Example: Optimisation

- ▶ Find a **set of numerical parameters** that optimises a function
 - ▶ The dimensions of a steel jacket of a wind turbine [BM21]
 - ▶ The weights and biases of a neural network
 - ▶ The hyper-parameters of an ML model to maximise accuracy [SSP+22]

- ▶ Find a **combination** of mathematical objects that optimises a function
 - ▶ A sequence of algorithms to apply
 - ▶ A permutation of storage depots to visit
 - ▶ A set of neurons and their connections [SCLM19]

Learning, control, and optimisation, can all be expressed as search problems!

Evolutionary Search

Searching in a box of states: \mathcal{X}

In a nutshell, an Evolutionary Algorithm (EA) starts with a population $P^t \subset \mathcal{X}$ of randomly generated states. Then:

1. Stochastically select *parent states* $P_\mu \subset P^t$ to recombine (**selection**)
2. Stochastically generate *offspring states* $P_\lambda \subset \mathcal{X}$ from parents (**crossover**)
3. Stochastically apply a perturbation to the offspring states P_λ (**mutation**)
4. Stochastically select the *survivors* for the next generation $P^{t+1} \subseteq P_\mu \cup P_\lambda$

This process will be repeated until a solution has been found or until enough generations have been replaced.

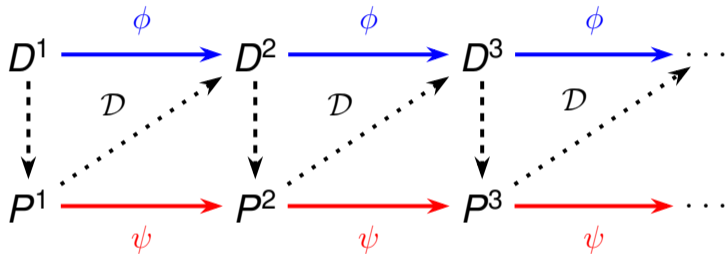
Section 2

Why are EAs *good* at search?

Abstraction: Probability Distributions Evolving

Evolutionary operators are sampling techniques in disguise

A population can be thought of as a probability distribution changing through time [Leh24].



From Lehre, P. K. (2024). Runtime Analysis of Population-based Evolutionary Algorithms. GECCO '24.

Transformations ψ on a population P^t results in a new population P^{t+1} .

All metaheuristics do this!

If all metaheuristics are just sampling the distributions through time,
why do we have so many different algorithms out there?

Exposing the ‘novel’ metaphors

Should we ban the word *novel* from research?



Article

Metaheuristics—the metaphor exposed

[Kenneth Sörensen](#)

First published: 08 February 2013 | <https://doi.org/10.1111/itor.12001> | [VIEW METRICS](#)



Volume 22, Issue 1

Special Issue:Matheuristics:
Model-Based Metaheuristics

January 2015

Pages 3-18

This article also appears in:
ITOR - Special Virtual 30th
Anniversary Issue

[Sör15]

Home > Swarm Intelligence > Article

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

Open access | Published: 30 November 2021
Volume 16, pages 1–6 (2022) [Cite this article](#)

You have full access to this open access article

[Download PDF](#) [Save article](#)

Claus Aranha, Christian L. Camacho Villalón, Felipe Campelo, Marco Dorigo, Rubén Ruiz, Marc Sevaux, Kenneth Sörensen & Thomas Stützle

22k Accesses 182 Citations 58 Altmetric 4 Mentions [Explore all metrics](#)

Sections

[References](#)

[Author information](#)

[ACC+22]

Computers & Operations Research 142 (2022) 105747

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/ator

An analysis of why cuckoo search does not bring any novel ideas to optimization

Christian L. Camacho-Villalón¹, Marco Dorigo, Thomas Stützle
¹IRISA, Université de Bretagne, Avenue Franklin Roosevelt 20, 35000, Rennes, Belgium

[CDS22]

There is even a ‘Pokédex’-like bestiary on GitHub! <https://github.com/fcampelo/EC-Bestuary> [CA18]

How do we guide the search, then?

And what are the pitfalls of metaheuristics?



*“Gradients are simple, cheap
(and amazingly accurate)
predictors of future states”*

We need to sample states that are
different!

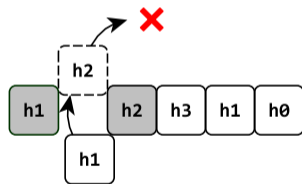
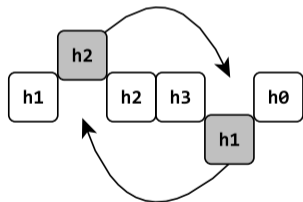
And difference will *hopefully* be achieved by changing something.

Mutation: Small Steps

The Stepping Stones of Evolution

Small perturbations on existing solutions create new states to **explore**¹

$$(0\ 1\ 1\ 0\ 0\ 1\ 0\ 0) \xrightarrow{mut} (0\ 1\ 1\ 0\ 0\ 1\ 0\ 1)$$



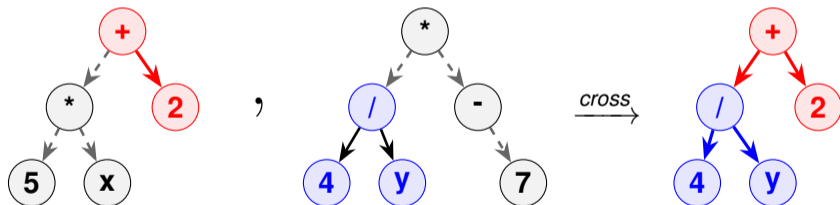
¹Exemplified here in discrete search spaces but generalisable to real spaces

Crossover: Large Jumps

The Stepping Stones of Evolution

Large perturbations **exploit** promising solutions creating new states to **explore**²

$$(1\ 1\ 0\ 0\ 1\ 1\ 0\ 0), (0\ 1\ 1\ 0\ 0\ 1\ 0\ 0) \xrightarrow{\text{cross}} (1\ 1\ 0\ 0\ 1\ 1\ 0\ 0)$$



²It can be argued that crossover is just mutation using a larger neighbourhood size!

Selection and Survival

The Stepping Stones of Evolution

Simply put, **selection** is just deciding on how to choose the *good* states.

- ▶ *Parent selection* determines which states get to be **generated**
- ▶ *Survival selection* determines which of the **generated** states will be resampled

The key lies on generating *good* states.³

³What does *good* mean?

Searching randomly is not enough

We **need** *might* benefit
from **context!**

Section 3

Problem Characterisation

Problem Characterisation

A smarter approach to search

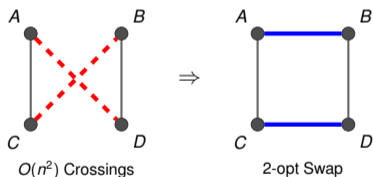
It is useful to possess some a priori knowledge of the problem at hand [KPWT16].

- ▶ **Domain knowledge:** problem-specific information that is useful for the search.
 - ▶ E.g., “these two depots are close, so probably visiting one after the other makes sense”
- ▶ **Instance-specific:** this specific instance has some *exploitable structure*.
 - ▶ E.g., “the search space *looks* like this, so I should probably move in that direction”

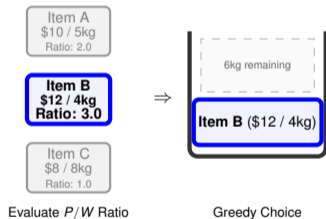
Problem-specific Operators

Using domain knowledge

Problem characterisation is useful when f is unknown [MBT+11], when we need to generalise over several instances [BHK+19], or when we want to create **problem-specific operators**.



TSP: Uncrossing local edges



Knapsack: Profit/weight as a heuristic

Exploratory Landscape Analysis

Using instance-specific information

Exploratory Landscape Analysis (ELA) is a way to find interactions between problem instance properties and algorithm performance [MBT+11]. Originally proposed in 2010 to assess BBOB'09 benchmarks [MPT10]. They looked at low-level properties resulting in the following high-level properties:

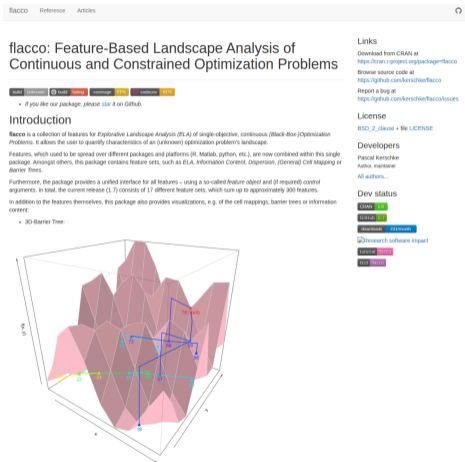
- ▶ Multimodality
- ▶ Global structure
- ▶ Separability
- ▶ Variable scaling
- ▶ Search space homogeneity
- ▶ Basin size homogeneity
- ▶ Global to local optima contrast
- ▶ Plateaus

ELA in Practice

Using instance-specific information

The *classic ELA features* included 50 different features. Today, many more have been developed and are conveniently packed in two main software packages: **flacco** [KT16] in R, and **pflacco** [PT23] in Python.

- ▶ A GUI is available for flacco [HK17]
- ▶ Statistics from both packages can be incorporated into IOH workflows [WVY+22]



flacco: Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems

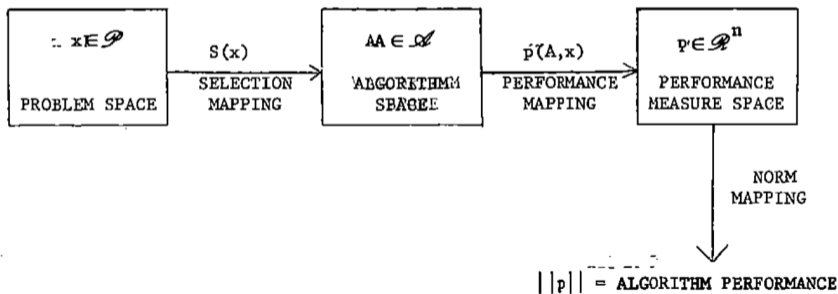
flacco is a collection of features for Explorative Landscape Analysis (ELA) of single-objective, continuous (black-box) Optimization Problems. It allows the user to quantify characteristics of an (unknown) optimization problems landscape. Features, which used to be spread over different packages and platforms (R, Matlab, python, etc.), are now combined within this single package. Amongst others, this package contains feature sets, such as EIA, Information Content, Dispersion, (Deneva) Cell Mapping or Barrier Trees. Furthermore, the package provides a unified interface for all features – using a so-called feature object and (if required) control arguments. In total, the current release (1.7) consists of 17 different feature sets, which sum up to approximately 300 features. In addition to the features themselves, this package also provides visualizations, e.g. of the cell mappings, barrier trees or information content.

3D Barrier Tree

The Algorithm Selection Problem

Using domain knowledge AND instance-specific information

- ▶ **Performance complementarity:** different instances are best solved using different algorithms [KHNT19]
- ▶ **The Algorithm Selection Problem (ASP):** Determine the appropriate solver for a given problem such that algorithm performance is maximised [Ric76]



But you can use both!

Using both domain knowledge AND instance-specific information is necessary for automatic algorithm design!

The ASP and Hyper-heuristics

Using domain knowledge AND instance-specific information

Hyper-heuristics (HH) are *algorithms to select or generate algorithms* [BHK+19].

- ▶ **Selection hyper-heuristics** search in the space of solvers to select an appropriate algorithm given the problem instance (or current problem state)
- ▶ **Generation hyper-heuristics** generate new heuristics or operators from primitive components

To determine which operator or component to choose, we need **problem characterisation!**

Some work in HHs

Adaptive problem characterisation using EAs

- ▶ **Master thesis** on selection HHs for 0/1 Knapsack and Constraint Satisfaction [Sán17]
- ▶ **WCCI 2020: A (1+1)-EA** used as a selection HH for 0/1 Knapsack [SOA+20]
- ▶ **MDPI Appl. Sci. 2021: HHs** outperform single operators even in hard instances [SOA+21]

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences



Analysis of a Feature-independent Hyper-heuristic Model for Constraint Satisfaction and Binary Knapsack Problems

A thesis presented by

Xavier Fernando Cuauhtémoc Sánchez Díaz

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Intelligent Systems

Monterrey, Nuevo León, December, 2017

A Preliminary Study on Feature-independent Hyper-heuristics for the 0/1 Knapsack Problem

Xavier F. C. Sánchez-Díaz*, José Carlos Ortiz-Baylón[†], Ivan Amaya[‡],
Jorge M. Cruz-Duarte[§], Santiago Enrique Cozatl-Pablos[¶], and Hugo Terashima-Marin[¶]
^{*} Tecnológico de Monterrey, School of Engineering and Sciences
E-mails: {xsc, jcozatl, ianaya, jpcruz, sencoat, terashima}@itesm.mx

Abstract—Recent years have witnessed an escalating interest for methods that automatically adapt to different types of problems. In this regard, the term hyper-heuristics—heuristics that either select or generate new heuristics—is a relevant concept. Experimental evidence supports the idea that hyper-heuristics can outperform single, isolated heuristics. However, consistent and hyper-heuristic models require several inputs. One of them is a set of features that accurately characterize the instances, which limits their applicability. Thus, in this work, we analyze how to implement a simple, evolutionary algorithm to produce feature-independent hyper-heuristics. We compare its performance against that of single heuristics, for the details of the knapsack problem. Our research focuses on two elements: performance and frequency. In the former, we analyze how the performance of the learning stage varies across different scenarios. In the latter, we examine how frequently heuristics interact within the hyper-heuristic. We show that the proposed hyper-heuristic model solves most of the instances considered in this work. Moreover, it does so more efficiently than isolated heuristics. At the same time, the model offers a straightforward parameter setting and requires little or no problem characterization, which simplifies its use on new problem domains.

Index Terms—Heuristics, Hyper-heuristics, Knapsack problem.

more acceptance determines whether the new solution is accepted or discarded. The approach proposed in this work simplifies the overall model by only focusing on heuristic selection. Thus, changes resulting from applying a particular heuristic are always accepted.

Among the many learning methods used in the literature to produce hyper-heuristics, evolutionary computation is a recurring one. Examples of these methods include, but are not limited to Genetic Programming (GP) [4], Grammatical Evolution (GE) [6], Many Genetic Algorithms (MGA) [7], and Artificial Immune Systems (AIS) [8], [9]. Even so, there are other proposals, such as those that create the hyper-heuristics by analyzing the set of problem instances [10].

Despite the wide use of hyper-heuristics [5], [11], only a few works explore the strengths of these heuristics. A few examples include the runtime analysis of selection hyper-heuristics [12], [13], the use and control of crossover operators for selection hyper-heuristics [14], and heuristic interaction of selection hyper-heuristics applied to constraint satisfaction problems [15]. Therefore, this paper conducts an exploration of heuristic interaction for the 0/1 knapsack problem using



Article

A Feature-Independent Hyper-Heuristic Approach for Solving the Knapsack Problem

Xavier Sánchez-Díaz , José Carlos Ortiz-Baylón , Ivan Amaya , Jorge M. Cruz-Duarte ,
Santiago Enrique Cozatl-Pablos  and Hugo Terashima-Marin 

Tecnológico de Monterrey, School of Engineering and Sciences, Monterrey 64600, Mexico
xsc@itesm.mx (X.F.C.S.-D.); jcozatl@itesm.mx (J.C.O.-B.); ianaya@itesm.mx (I.A.); jpcruz@itesm.mx (J.M.C.-D.); sencoat@itesm.mx (S.E.C.-P.);
terashima@itesm.mx (H.T.-M.)
^{*} Correspondence: jcozatl@itesm.mx

Abstract: Recent years have witnessed a growing interest in automatic learning mechanisms and applications. The concept of hyper-heuristics, algorithms that either select among existing algorithms or generate new ones, holds high relevance in this matter. Current research suggests that, under certain circumstances, hyper-heuristics outperform single heuristics when evaluated in isolation. When hyper-heuristics are selected among existing algorithms, they may problem states into suitable subsets. Unfortunately, identifying the heuristics that accurately describe the problem state—and thus allow for a proper mapping—requires plenty of domain-specific knowledge, which is not always available. This work proposes a simple yet effective hyper-heuristic model that does not rely on problem features to produce such a mapping. The model defines a fixed sequence of heuristics that improves the solving process of knapsack problems. This research compares an analysis of feature-independent hyper-heuristic performance under different learning conditions and diverse problem sets.

Keywords: hyper-heuristics; knapsack problem; optimization



Citation: Sánchez-Díaz, X., Ortiz-Baylón, J.C., Amaya, I., Cruz-Duarte, J.M., Cozatl-Pablos, S.E., Terashima-Marin, H. A Feature-Independent Hyper-Heuristic Approach for Solving the Knapsack Problem. *Applied Sciences* 2021, 11, 11111. <https://doi.org/10.3390/app112111111>

Section 4

Challenges in Search

Challenges in Search

Besides expensive computations, what else could go wrong?

Many factors influence algorithm behaviour, so things can go awry even with problem characterisation.

- ▶ **Deceptiveness**⁴
 - ▶ When the search landscape is *deceiving*
- ▶ **Linkage** (or Epistasis)⁴
 - ▶ When several problem dimensions are *linked*: related and dependent on each other [PKF21]
- ▶ **Noise**⁴
 - ▶ When there is uncertainty in the fitness evaluation [Sud18]

⁴All of these things happen in real-world applications!

Challenges in Search

Besides expensive computations, what else could go wrong?

Many factors influence algorithm behaviour, so things can go awry even with problem characterisation.

- ▶ **Deceptiveness**⁴
 - ▶ When the **search landscape** is *deceiving*
- ▶ **Linkage** (or Epistasis)⁴
 - ▶ When several problem dimensions are *linked*: related and dependent on each other [PKF21]
- ▶ **Noise**⁴
 - ▶ When there is uncertainty in the fitness evaluation [Sud18]

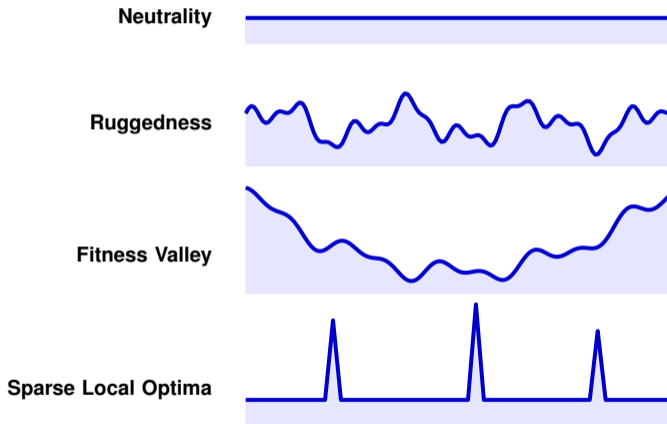
⁴All of these things happen in real-world applications!

Landscape Structure

A deep dive into landscape analysis

Several *structures* can be identified if we were to (somehow) plot the fitness function [Ree14; MO21].

Algorithms and samplers (e.g. for **ELA**) need to traverse this landscape, which can be *tricky*.



Dense Valleys and Sparse Optima

Balancing the search

We have a hunch—dense fitness valleys are easy to navigate, while sparse local optima are difficult to reach.

However, recent advances have been made to quantify exactly how difficult a problem⁵ might be depending on the amount of dense valleys and sparse optima in the search space [[Leh24](#); [DL26](#)].

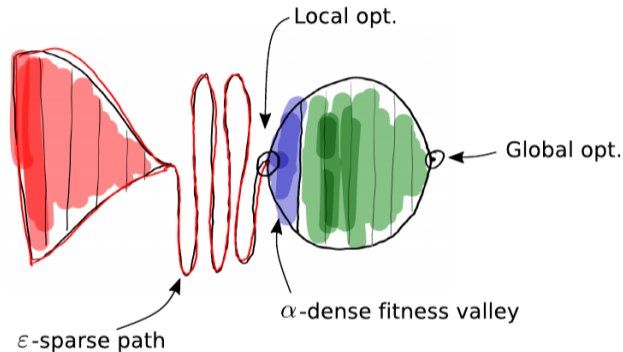
⁵in the pseudo-Boolean setting

A path is needed

An example from theory: BBFunnel

Different navigation tactics are needed at different stages of the search.

- ▶ **Red** areas reward exploration
- ▶ **Blue** areas reward exploitation
- ▶ **Green** areas are monotonic



BBFunel [DEL21], from Lehre, P.K. 2024. Runtime Analysis of Population-based Evolutionary Algorithms. GECCO 2024

A reachable path is needed

Theory: Level-based partition for pseudo-Boolean functions

Assume that we have a sampling process that has positive drift towards our goal [Köt24]. Then, the search space can be partitioned into partially ordered levels $A_1 \dots A_m$ [Leh11] leading to our goal. If:

1. The number of levels m is not too large: $m \in \text{poly}(n)$,
2. The distance between two adjacent levels is short: $O(1)$, and
3. The last level A_m contains all optimal points,

then we can look for deceptive pairs, and see if local optima are ϵ -sparse and fitness valleys are α -dense to determine the problem difficulty.

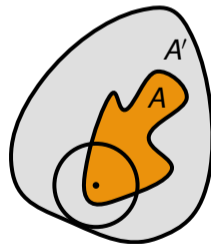
α -density

More theory: $SLO_{\epsilon,r}^\alpha$ -hierarchy

Informal definition of α -density (from [DL26])

A subset C is α -dense with respect to D if at least a fraction $\alpha \in [0, 1]$ of its neighbours are in D .

A' is α -dense with respect to A because at least α -neighbors are in A'



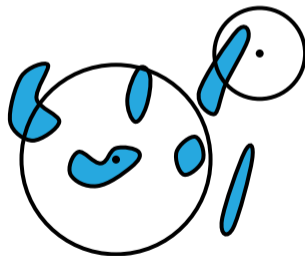
ϵ, r -sparsity

More theory: $SLO_{\epsilon, r}^\alpha$ -hierarchy

Informal definition of ϵ, r_{\max} -sparsity (from [DL26])

A subset B is (ϵ, r_{\max}) -sparse if it contains at most a fraction $\epsilon \in [0, 1]$ of all neighbours (in all radiuses $r_1 \dots r_{\max} \in [n - 1]$) of all the states in the search space.

B is ϵ, r_{\max} -sparse because it has at least a fraction ϵ of all states that are reachable in $O(\frac{1}{n} \binom{n}{r})$ for every neighbourhood of size $r \in [n - 1]$.



Sparse optima create deceptiveness

More theory: $SLO_{\epsilon,r}^\alpha$ -hierarchy

Informal implications of α -density and ϵ, r -sparsity

- ▶ **Good**: if most of the neighbours of the current state lead to a *better level*
- ▶ **Not good**: if there are reachable areas from the current state that have better fitness but do not lead to a *better level*
 - ▶ Formally: a pair (A_i, A_j) is f -deceptive if $1 \leq i < j \leq m$ and $f(A_i) \geq f(A_j)$

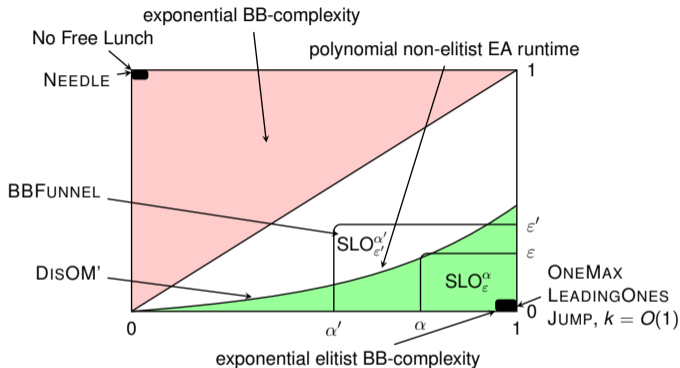
Algorithms need to **sample from better levels**, which can be tricky if there are many spikes that *appear* better but are not!

The $SLO_{\epsilon, r}^{\alpha}$ -hierarchy for Pseudo-Boolean Functions

A good summary

Depending on the ratio of α and ϵ , a hierarchy of problem complexity can be made.^a

- ▶ Short version from GECCO 2024 [DL24].
- ▶ Extended version from Algorithmica, 2026 [DL26].



^aAppropriate parameter setting is still needed!

From Dang, D.-C., & Lehre, P. K. (2026). The SLO Hierarchy of Pseudo-Boolean Functions and Runtime of Evolutionary Algorithms. *Algorithmica*, 88(2), 32.

Implications for Search and Optimisation

Wrapping-up

- ▶ We have a rough idea⁶ of how search works
- ▶ Domain knowledge helps us guide the search
- ▶ Instance-specific information helps identify problem structure
- ▶ However, the real world is way more nuanced!
 - ▶ Our guess could be very wrong if the problem is **deceptive**, **dynamic**, or **noisy**
(or **uncertain**, or **constrained**, or **multimodal**, ...)
 - ▶ Applied research needs as much problem characterisation as possible

⁶more like an educated guess

Before applying an algorithm, ask yourself:

Is there even something to learn?
(or some structure to exploit?)

Probably there is. Just look at the space :)

References I

- [ACC+22] C. Aranha et al., “Metaphor-based metaheuristics, a call for action: The elephant in the room,” *Swarm Intelligence*, vol. 16, no. 1, pp. 1–6, Mar. 1, 2022, ISSN: 1935-3820. DOI: [10.1007/s11721-021-00202-9](https://doi.org/10.1007/s11721-021-00202-9) Accessed: Aug. 18, 2022. [Online]. Available: <https://doi.org/10.1007/s11721-021-00202-9>
- [BM21] J. Burak and O. J. Mengshoel, “A multi-objective genetic algorithm for jacket optimization,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '21, New York, NY, USA: Association for Computing Machinery, Jul. 8, 2021, pp. 1549–1556, ISBN: 978-1-4503-8351-6. DOI: [10.1145/3449726.3463150](https://dl.acm.org/doi/10.1145/3449726.3463150) Accessed: Sep. 1, 2023. [Online]. Available: <https://dl.acm.org/doi/10.1145/3449726.3463150>
- [BHK+19] E. K. Burke, M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, “A Classification of Hyper-Heuristic Approaches: Revisited,” in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds., Cham: Springer International Publishing, 2019, pp. 453–477, ISBN: 978-3-319-91086-4. DOI: [10.1007/978-3-319-91086-4_14](https://doi.org/10.1007/978-3-319-91086-4_14) Accessed: Jun. 3, 2026. [Online]. Available: https://doi.org/10.1007/978-3-319-91086-4_14
- [CDS22] C. L. Camacho-Villalón, M. Dorigo, and T. Stützle, “An analysis of why cuckoo search does not bring any novel ideas to optimization,” *Computers & Operations Research*, vol. 142, p. 105747, Jun. 1, 2022, ISSN: 0305-0548. DOI: [10.1016/j.cor.2022.105747](https://www.sciencedirect.com/science/article/pii/S0305054822000442) Accessed: Nov. 18, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054822000442>

References II

- [CA18] F. Campelo and C. Aranha, “EC Bestiary: A bestiary of evolutionary, swarm and other metaphor-based algorithms,” Jun. 20, 2018. DOI: [10.5281/zenodo.1293352](https://doi.org/10.5281/zenodo.1293352) Accessed: Jun. 2, 2026. [Online]. Available: <https://zenodo.org/records/1293352>
- [DEL21] D.-C. Dang, A. Eremeev, and P. K. Lehre, “Non-elitist evolutionary algorithms excel in fitness landscapes with sparse deceptive regions and dense valleys,” in Proceedings of the Genetic and Evolutionary Computation Conference, ser. GECCO '21, New York, NY, USA: Association for Computing Machinery, Jun. 26, 2021, pp. 1133–1141, ISBN: 978-1-4503-8350-9. DOI: [10.1145/3449639.3459398](https://doi.org/10.1145/3449639.3459398) Accessed: Jul. 13, 2022. [Online]. Available: <https://doi.org/10.1145/3449639.3459398>
- [DL24] D.-C. Dang and P. K. Lehre, “The SLO Hierarchy of pseudo-Boolean Functions and Runtime of Evolutionary Algorithms,” in Proceedings of the Genetic and Evolutionary Computation Conference, ser. GECCO '24, New York, NY, USA: Association for Computing Machinery, Jul. 14, 2024, pp. 1551–1559, ISBN: 979-8-4007-0494-9. DOI: [10.1145/3638529.3654221](https://doi.org/10.1145/3638529.3654221) Accessed: Oct. 23, 2024. [Online]. Available: <https://dl.acm.org/doi/10.1145/3638529.3654221>
- [DL26] D.-C. Dang and P. K. Lehre, “The SLO Hierarchy of Pseudo-Boolean Functions and Runtime of Evolutionary Algorithms,” Algorithmica, vol. 88, no. 2, p. 32, Mar. 24, 2026, ISSN: 1432-0541. DOI: [10.1007/s00453-025-01359-z](https://doi.org/10.1007/s00453-025-01359-z) Accessed: May 8, 2026. [Online]. Available: <https://doi.org/10.1007/s00453-025-01359-z>

References III

- [Dow25] K. L. Downing, “Prediction at the core of both natural and artificial intelligence,” presented at the Norwegian AI Society Symposium (Tromsø, Norway), Jun. 19, 2025.
- [HK17] C. Hanster and P. Kerschke, “Flaccogui: Exploratory landscape analysis for everyone,” in Proceedings of the Genetic and Evolutionary Computation Conference Companion, ser. GECCO ’17, New York, NY, USA: Association for Computing Machinery, Jul. 15, 2017, pp. 1215–1222, ISBN: 978-1-4503-4939-0. DOI: [10.1145/3067695.3082477](https://doi.org/10.1145/3067695.3082477) Accessed: Jun. 4, 2026. [Online]. Available: <https://dl.acm.org/doi/10.1145/3067695.3082477>
- [KHNT19] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, “Automated Algorithm Selection: Survey and Perspectives,” Evolutionary Computation, vol. 27, no. 1, pp. 3–45, Mar. 1, 2019, ISSN: 1063-6560. DOI: [10.1162/evco_a_00242](https://doi.org/10.1162/evco_a_00242) Accessed: Jun. 4, 2026. [Online]. Available: https://doi.org/10.1162/evco_a_00242
- [KPWT16] P. Kerschke, M. Preuss, S. Wessing, and H. Trautmann, “Low-Budget Exploratory Landscape Analysis on Multiple Peaks Models,” in Proceedings of the Genetic and Evolutionary Computation Conference 2016, ser. GECCO ’16, New York, NY, USA: Association for Computing Machinery, Jul. 20, 2016, pp. 229–236, ISBN: 978-1-4503-4206-3. DOI: [10.1145/2908812.2908845](https://doi.org/10.1145/2908812.2908845) Accessed: Apr. 10, 2024. [Online]. Available: <https://dl.acm.org/doi/10.1145/2908812.2908845>

References IV

- [KT16] P. Kerschke and H. Trautmann, “The R-Package FLACCO for exploratory landscape analysis with applications to multi-objective optimization problems,” in 2016 IEEE Congress on Evolutionary Computation (CEC), Jul. 2016, pp. 5262–5269. DOI: [10.1109/CEC.2016.7748359](https://doi.org/10.1109/CEC.2016.7748359) Accessed: Jun. 4, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/7748359>
- [Köt24] T. Kötzing, “Theory of Stochastic Drift,” arXiv: [2406.14589](https://arxiv.org/abs/2406.14589) [math], Accessed: Oct. 2, 2025. [Online]. Available: <http://arxiv.org/abs/2406.14589> pre-published.
- [Leh11] P. K. Lehre, “Fitness-levels for non-elitist populations,” in Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, ser. GECCO ’11, New York, NY, USA: Association for Computing Machinery, Jul. 12, 2011, pp. 2075–2082, ISBN: 978-1-4503-0557-0. DOI: [10.1145/2001576.2001855](https://doi.org/10.1145/2001576.2001855) Accessed: Jun. 8, 2026. [Online]. Available: <https://dl.acm.org/doi/10.1145/2001576.2001855>
- [Leh24] P. K. Lehre, “Runtime Analysis of Population-based Evolutionary Algorithms,” in Proceedings of the Genetic and Evolutionary Computation Conference Companion, ser. GECCO ’24 Companion, New York, NY, USA: Association for Computing Machinery, Aug. 1, 2024, pp. 903–927, ISBN: 979-8-4007-0495-6. DOI: [10.1145/3638530.3648428](https://doi.org/10.1145/3638530.3648428) Accessed: Mar. 1, 2025. [Online]. Available: <https://dl.acm.org/doi/10.1145/3638530.3648428>

References V

- [MO21] K. Malan and G. Ochoa, “Recent advances in landscape analysis for optimisation and learning,” in Proceedings of the Genetic and Evolutionary Computation Conference Companion, ser. GECCO '21, New York, NY, USA: Association for Computing Machinery, Jul. 8, 2021, pp. 899–917, ISBN: 978-1-4503-8351-6. DOI: [10.1145/3449726.3461396](https://doi.org/10.1145/3449726.3461396) Accessed: Jan. 27, 2024. [Online]. Available: <https://dl.acm.org/doi/10.1145/3449726.3461396>
- [MBT+11] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, “Exploratory landscape analysis,” in Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, ser. GECCO '11, New York, NY, USA: Association for Computing Machinery, Jul. 12, 2011, pp. 829–836, ISBN: 978-1-4503-0557-0. DOI: [10.1145/2001576.2001690](https://doi.org/10.1145/2001576.2001690) Accessed: Jan. 22, 2024. [Online]. Available: <https://dl.acm.org/doi/10.1145/2001576.2001690>
- [MPT10] O. Mersmann, M. Preuss, and H. Trautmann, “Benchmarking Evolutionary Algorithms: Towards Exploratory Landscape Analysis,” in Parallel Problem Solving from Nature, PPSN XI, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds., Berlin, Heidelberg: Springer, 2010, pp. 73–82, ISBN: 978-3-642-15844-5. DOI: [10.1007/978-3-642-15844-5_8](https://doi.org/10.1007/978-3-642-15844-5_8)
- [PT23] R. P. Prager and H. Trautmann, “Pflacco: Feature-based landscape analysis of continuous and constrained optimization problems in python,” Evolutionary Computation, pp. 1–25, Jul. 2023, ISSN: 1063-6560. DOI: [10.1162/evco_a_00341](https://doi.org/10.1162/evco_a_00341) eprint: https://direct.mit.edu/evco/article-pdf/doi/10.1162/evco_a_00341/2148122/evco_a_00341.pdf. [Online]. Available: https://doi.org/10.1162/evco_a_00341

References VI

- [PKF21] M. W. Przewozniczek, M. M. Komarnicki, and B. Frej, “Direct linkage discovery with empirical linkage learning,” in Proceedings of the Genetic and Evolutionary Computation Conference, ser. GECCO '21, New York, NY, USA: Association for Computing Machinery, Jun. 26, 2021, pp. 609–617, ISBN: 978-1-4503-8350-9. DOI: [10.1145/3449639.3459333](https://doi.org/10.1145/3449639.3459333) Accessed: Jun. 4, 2026. [Online]. Available: <https://dl.acm.org/doi/10.1145/3449639.3459333>
- [Ree14] C. R. Reeves, “Fitness Landscapes,” in Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, E. K. Burke and G. Kendall, Eds., Boston, MA: Springer US, 2014, pp. 681–705, ISBN: 978-1-4614-6940-7. DOI: [10.1007/978-1-4614-6940-7_22](https://doi.org/10.1007/978-1-4614-6940-7_22) Accessed: Jan. 28, 2024. [Online]. Available: https://doi.org/10.1007/978-1-4614-6940-7_22
- [Ric76] J. R. Rice, “The Algorithm Selection Problem,” in Advances in Computers, vol. 15, Elsevier, Jan. 1, 1976, pp. 65–118. DOI: [10.1016/S0065-2458\(08\)60520-3](https://www.sciencedirect.com/science/chapter/bookseries/abs/pii/S0065245808605203) Accessed: Jun. 4, 2026. [Online]. Available: <https://www.sciencedirect.com/science/chapter/bookseries/abs/pii/S0065245808605203>
- [SOA+21] X. Sánchez-Díaz, J. C. Ortiz-Bayliss, I. Amaya, J. M. Cruz-Duarte, S. E. Conant-Pablos, and H. Terashima-Marín, “A Feature-Independent Hyper-Heuristic Approach for Solving the Knapsack Problem,” Applied Sciences, vol. 11, no. 21, p. 10 209, 21 Jan. 2021, ISSN: 2076-3417. DOI: [10.3390/app112110209](https://www.mdpi.com/2076-3417/11/21/10209) Accessed: Nov. 18, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/11/21/10209>

References VII

- [Sán17] X. F. C. Sánchez-Díaz, “Analysis of a Feature-independent Hyper-heuristic Model for Constraint Satisfaction and Binary Knapsack Problems,” M.S. thesis, Tecnológico de Monterrey, Monterrey, Mexico, Dec. 2017, 88 pp. Accessed: Jan. 30, 2026. [Online]. Available: https://saxarona.github.io/assets/pdf/master_thesis.pdf
- [SOA+20] X. F. C. Sánchez-Díaz, J. C. Ortiz-Bayliss, I. Amaya, J. M. Cruz-Duarte, S. E. Conant-Pablos, and H. Terashima-Marín, “A Preliminary Study on Feature-independent Hyper-heuristics for the 0/1 Knapsack Problem,” in *2020 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2020, pp. 1–8. DOI: [10.1109/CEC48606.2020.9185671](https://doi.org/10.1109/CEC48606.2020.9185671) Accessed: Nov. 30, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/9185671>
- [SSP+22] L. Schneider, L. Schäpermeier, R. P. Prager, B. Bischl, H. Trautmann, and P. Kerschke, “HPO \times ELA: Investigating Hyperparameter Optimization Landscapes by Means of Exploratory Landscape Analysis,” in *Parallel Problem Solving from Nature — PPSN XVII*, G. Rudolph, A. V. Kononova, H. Aguirre, P. Kerschke, G. Ochoa, and T. Tušar, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2022, pp. 575–589, ISBN: 978-3-031-14714-2. DOI: [10.1007/978-3-031-14714-2_40](https://doi.org/10.1007/978-3-031-14714-2_40)
- [Sör15] K. Sörensen, “Metaheuristics—the metaphor exposed,” *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015, ISSN: 1475-3995. DOI: [10.1111/itor.12001](https://doi.org/10.1111/itor.12001) Accessed: Jun. 2, 2026. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12001>

References VIII

- [SCLM19] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, “Designing neural networks through neuroevolution,” *Nature Machine Intelligence*, vol. 1, no. 1, pp. 24–35, 1 Jan. 2019, ISSN: 2522-5839. DOI: [10.1038/s42256-018-0006-z](https://doi.org/10.1038/s42256-018-0006-z) Accessed: Sep. 27, 2022. [Online]. Available: <https://www.nature.com/articles/s42256-018-0006-z>
- [Sud18] D. Sudholt, “On the robustness of evolutionary algorithms to noise: Refined results and an example where noise helps,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '18, New York, NY, USA: Association for Computing Machinery, Jul. 2, 2018, pp. 1523–1530, ISBN: 978-1-4503-5618-3. DOI: [10.1145/3205455.3205595](https://doi.org/10.1145/3205455.3205595) Accessed: Jun. 4, 2026. [Online]. Available: <https://dl.acm.org/doi/10.1145/3205455.3205595>
- [WVY+22] H. Wang, D. Vermetten, F. Ye, C. Doerr, and T. Bäck, “IOAnalyzer: Detailed Performance Analyses for Iterative Optimization Heuristics,” *ACM Transactions on Evolutionary Learning and Optimization*, vol. 2, no. 1, 3:1–3:29, Apr. 5, 2022, ISSN: 2688-299X. DOI: [10.1145/3510426](https://doi.org/10.1145/3510426) Accessed: Jun. 1, 2026. [Online]. Available: <https://dl.acm.org/doi/10.1145/3510426>