Education & Technology

Tommaso Minerva | Annamaria De Santis

# GENETIC ALGORITHMS AND EVOLUTIONARY COMPUTATION IN EDUCATION

Forewords by
Pier Cesare Rivoltella | Marcello Chiodi

Pearson

EDUNEXT
Italian Universities Digital Education Hub

Tommaso Minerva | Annamaria De Santis

# GENETIC ALGORITHMS AND EVOLUTIONARY COMPUTATION IN EDUCATION

Forewords by
Pier Cesare Rivoltella | Marcello Chiodi

Pearson

EDUNEXT
Italian Universities Digital Education Hub

*Our authorised representative in the EU for product safety is Pearson Benelux B.V., Kabelweg 37, 1014 BA Amsterdam, Netherlands, productsafety@pearson.com.*



Pearson è sensibile e attenta ai temi dell'inclusione e della diversity; siamo ovviamente aperti ai vostri suggerimenti su come progredire in questo percorso, per costruire esperienze di apprendimento libere da pregiudizi e inclusive. Contattateci su inclusione@pearson.com

Grafica di copertina: Maurizio Garofalo
Immagine di copertina: Alx/stock.adobe.com
Stampa: Lego Digit - Lavis (TN)



Member of CISQ Federation
CERTIFIED MANAGEMENT SYSTEM
ISO 9001

9788891945464

1ª edizione: ottobre 2026

Ristampa
00 01 02 03 04

Anno
26 27 28 29 30

# Contents

# The authors

**Tommaso MINERVA** is a Full Professor of Statistics at the University of Modena and Reggio Emilia. Holding a Ph.D. in Solid State Physics, his early research focused on computational simulations of the electronic structure of superconducting oxides. Subsequently, his interests shifted towards computational techniques for data analysis, and since the late 1990s, he has been involved in biology-inspired algorithms (genetic and evolutionary algorithms, artificial neural networks, fuzzy logic), which are precursors to modern artificial intelligence algorithms. More recently, his research interests have turned to the application of statistical-computational approaches in the field of education and learning analytics. Tommaso Minerva is the Scientific Coordinator of the EDUNEXT project.

**Annamaria DE SANTIS** is a Tenure-Track Researcher in Experimental Pedagogy at the University of Modena and Reggio Emilia. She holds a Ph.D. in Pedagogy and Educational Sciences and worked as an Instructional Designer at the Edunova Centre (2015-2023). Managing Editor of the *Journal of e-Learning and Knowledge Society*, she authored the book "Analisi multivariata e learning analytics: metodi e applicazioni" (Multivariate Analysis and Learning Analytics: Methods and Applications). Her teaching and research interests focus on the application of quantitative approaches to educational research and on the design of digital educational programmes, environments, and solutions. She directs the Master's in Digital Education Design and is part of the EDUNEXT coordination team.

## Acknowledgments

The authors acknowledge the financial support by EDUNEXT (Next Education Italia) project (CUP: E83C23003480007) through Italy's National Recovery and Resilience Plan (NRRP), Mission 4 "Education and Research" – Component 1 "Strengthening the provision of education services: from early childhood to universities" – Investment 3.4 "Advanced university teaching and skills" – Sub-investment 3 "Digital Education Hubs (DEH)", funded by the European Union – NextGenerationEU (MUR Director's Decree No. 2100, 15 December 2023) by the Minister of University and Research (MUR).

## AI Statement

During the preparation of this work, the authors used:

*Grammarly* to improve text readability and grammar check. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

*Scopus/Co-Pilot* to select references. After using this tool/service, authors checked, selected, and downloaded all the papers from the original journals and took full responsibility for references and citations.

*ChatGPT, Model 5 Pro* to create the companion software package, geneticaR, and example scripts in R. After using this tool/service, authors checked all the scripts and take full responsibility for them.

## Attributions

The book represents the result of the joint work of the authors, who collaborated throughout all phases of the research.

According to CRediT system:

- *Tommaso Minerva:* Conceptualization, Methodology, Investigation, Project administration, Software, Supervision, Visualization, Writing - original draft, Writing - review & editing (Chapters: Introduction, 1-9, 18).
- *Annamaria De Santis:* Conceptualization, Resources, Visualization, Writing - original draft (Chapters: 10-17), Writing - review & editing.

# Foreword by Pier Cesare Rivoltella

In education, whenever we try to introduce a new tool – whether pedagogical or methodological, as in the case of this work that I have the immense pleasure of presenting –, something more profound than the simple adoption of a technology occurs. What happens is that we are forced to change the way we think: the tool shapes our brain frame, as de Kerckhove (1991) says about the media. This means that the promise of innovation is never purely functional: it is cognitive, cultural, and sometimes even political. For this reason, when we approach evolutionary algorithms and computation inspired by natural processes, we cannot limit ourselves to considering them as just another technique for optimizing complex structures. We must ask ourselves how their logic – a logic of populations, variation, adaptation, selection – can transform the way we look at educational processes.

The merit of this book by Tommaso Minerva and Annamaria De Santis is that it addresses this issue without taking shortcuts. They do this in two ways. On the one hand, they bring to the training of teachers and researchers a set of methods – Genetic Algorithms, Evolutionary Computation – that belong to a piece of the history of Artificial Intelligence (AI) that is often overshadowed today by the power of generative models. On the other hand, they show how these tools can become design devices for schools and universities: not simply techniques for finding solutions, but real ways to make constraints visible, clarify objectives, and negotiate the compromises that every educational choice entails.

At a time when AI seems to be moving primarily in the direction of automation and replacement, returning to an evolutionary paradigm is healthy. It reminds us that educating does not mean seeking the optimal solution, but accompanying a trajectory (Biesta, 2017). That learning is an adaptive process, never linear, always situated (Terrenghi et al., 2019). That the complexity of educational institutions cannot be reduced to a calculation but must be explored through models that respect its plural, multi-objective, imperfect nature.

At this level, the book's contribution is eminently pedagogical. Talking about optimization in education here does not mean adopting an efficiency-driven logic; on the contrary, it means taking optimization as a design practice, as a way of making values and priorities explicit, of making the effects of decisions transparent. This is what I have called, in other contexts, algorithmic pedagogy (Panciroli & Rivoltella, 2024): not blind reliance on machines, but critical work on their functioning to guide their use within an educational project.

Minerva and De Santis offer us a map and, at the same time, a toolbox. The map is a clear, rigorous, accessible reconstruction of the fundamentals of evolutionary computation. The toolbox is the translation of those fundamentals into concrete problems: time tabling, personalization, resource allocation, early risk identification. But the real strength of the volume lies in its ability to hold these two levels together, showing how every technical application is always the result of meaningful work.

At a time when LLMs and generative systems seem to dominate the scene, this book invites a different stance: one that considers AI not as a monolithic block but as an ecosystem of techniques, each with its own history, its own domain of validity, its own implicit ethics (Bula & Niedzielski, 2021). Genetic Algorithms, read in this way, become a lens for understanding AI as a whole: not only as computation, but as a search for balance between constraints, as a space of possibilities to be explored. It is an approach that restores to schools and, more generally, to educational systems what is theirs: the ability to deal with uncertainty, to work by approximation, to negotiate solutions that are never definitive. At the same time, it invites education designers to take the technical dimension seriously, not to suffer it but to inhabit it critically.

For all these reasons, the work of the two authors comes at the right time. It helps those involved in education to understand a part of the research into the development and application of AI that deserves to be rediscovered. It offers researchers and teachers tools to design better, more consciously. And it reminds us of all that technology, when well designed, does not simplify the world: it makes it more readable.

*Pier Cesare Rivoltella*
*Bologna, 2025*

## References

Biesta, G. (2017). *The Rediscovery of Teaching*. London: Taylor & Francis.

De Kerckhove, D. (1991). *Brainframes: technology, mind and business*. Baarn: Bosch & Keuning.

Panciroli, C., & Rivoltella, P. C. (2024). Collaborating with Machines. AI, Literacies, School. *Journal of Education and Cultural Studies*, *1*, 17-48.

Terrenghi, I., Diana, B., Zurloni, V., Rivoltella, P. C., Elia, M., Castaner, M., Camerino, O., & Anguera, M. T. (2019). Episode of situated learning to enhance student engagement and promote deep learning: Preliminary results in a high school classroom. *Frontiers of Psychology*, *10* (JUN), 1415-N/A.

Bula, P., & Niedzielski, B. (2021). *Management, Organizations and Artificial Intelligence: Where Theory Meets Practice*. London: Routledge.

**Pier Cesare Rivoltella** is a full professor of Media and Education Technology at the University of Bologna, where he leads the Master's program MED. Media, Education, and Teaching for Digital Innovation.
He is the editor-in-chief of the journal REM – Research on Education and Media and co-editor of Scholè, Rivista di educazione e studi culturali.
He is a member of the scientific committee of several journals in Italy and abroad.
He is coordinator of the PhD School in Learning Sciences and Digital Technologies.
He teaches courses at various Italian and foreign universities.

# Foreword by Marcello Chiodi

The text by Tommaso Minerva and Annamaria De Santis has the great merit of covering an important, highly specialized topic in a very clear manner and at different reading levels: for specialists in quantitative methods, for experts in teaching and teaching methodology, but also for those approaching the fascinating subject of evolutionary algorithms for the first time and curious to see its applications in fields that are very familiar to us academics, following the pioneering proposals of the 1960s (Holland, 1975).

Today, GA represent an important topic, together with many implementations of artificial intelligence and data science, whenever dealing with new problems, with LLMs (Large Language Models), and whenever non-classical and non-standard numerical analysis algorithms must be used. Regarding the relationship between GAs and LLMs, the ring diagram presented in the first pages is particularly illuminating.

The analysis of Genetic Algorithms, inspired by their biological and evolutionary progenitors (such as mutation, selection and crossover), together with the discussion of selection mechanisms and fitness functions, is interesting and well explained.

The text rightly emphasizes that in many contexts, techniques based on gradients, faster paths and rapid descents, although fascinating and full of efficient implementations even for very large problems, are not suitable for all situations: not all problems can be formalized in a classical analytical way, not all problems are derivable, many problems are simply combinatorial; the continuity of the functions to be maximized is a pipe dream in many situations (Goldberg, 1989).

I grew up with numerical algorithms, whose beauty and charm I appreciated, but I honestly envy colleagues such as Tommaso and other pioneers in Italy who have used and proposed these fascinating algorithms inspired by nature, evolution, natural selection, cells, migrations in some cases, and many other less formal phenomena. At the time of my training, I would never have thought that ants (with ant colony optimization) and neurons (with neural networks) could be found in optimization and data analysis algorithms, even though they may not be directly part of the family of evolutionary algorithms but are nevertheless based on the functioning of nature or the human body.

Annamaria and Tommaso are continuing and giving substance to a line of research that was only pioneering in the field of statistics a few decades ago, applying it to their areas of expertise, which are not easily lent to classical numerical analysis.

For several decades, Genetic Algorithms have been gaining ground in certain contexts, and it is honestly encouraging to see this very interesting form of application in a field that, like algorithms, has undergone enormous evolution in recent times, namely the field of education with all the techniques of synchronous and asynchronous distance learning, MOOCs and all the other gadgets we use in our work.

The section at the end of the text on the integration of Genetic Algorithms and artificial intelligence techniques for the manipulation of big data is very interesting. In fact, the text repeatedly emphasizes the fact that nowadays in the field of education, for example with the use of MOOCs, professionals must deal with impressive amounts of data (just think of the time and methods of accessing materials, courses and sequences of topics etc.), and the integration of techniques certainly yields positive results. Whenever textual data is involved, inevitably, whatever the LLM, the amount of data grows enormously.

The considerations on the fact that traditional models rarely consider aspects related to socio-economic status, disability or linguistic background also have a significant impact. Evolutionary algorithms often allow us to identify the combinations of factors that affect outcomes and can allow us to directly insert constraints related to equity into optimization processes, generally allowing for a more inclusive distribution of resources in some educational contexts.

In conclusion, I believe that not only the community of statisticians and educators, but the entire community, academic and otherwise, can draw inspiration from this modern, clear and highly informative text.

*Marcello Chiodi*
*Palermo, 2025*

## References

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. AddisonWesley Publishing Company, Inc.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The University of Michigan Press.

**Marcello Chiodi** is a Full Professor of Statistics at the University of Palermo, where he has served since 2000. He has been a member of the Academic Senate and Director of the Department of Economic, Business, and Statistical Sciences from 2015 to 2018, and previously Coordinator of the Ph.D. program.
Since 2024, he is the President of the Italian Statistical Society (SIS - Società Italiana di Statistica).
His scientific interests span methodological statistics, multivariate statistics, and computational statistics, with applications in various fields. In recent years, his research has focused primarily on space-time point processes.

# GENETIC ALGORITHMS, EVOLUTIONARY COMPUTATION, LEARNING ANALYTICS, LARGE LANGUAGE MODELS BUNDLED IN EDUCATION

## Why Genetic Algorithms matter for Learning Analytics in the era of LLM

Modern AIs/LLMs are, fundamentally, a story about search and optimization. Whether we build a predictor, design a schedule, sequence a curriculum, allocate scarce resources, or choose which signals to measure, we are navigating vast decision spaces with uncertainty and constraints. Many of these spaces are jagged (non-convex), partly discrete, simulator-defined, or multi-objective. In this terrain, Evolutionary Computation (EC) – and especially Genetic Algorithms (GAs) – remains essential.

GAs approach optimization as probabilistic search over a population of candidate solutions (Holland, 1975, 1992; Goldberg, 1989; Eiben & Smith, 2015). A fitness function scores candidates; the best are selected, their structure is recombined (crossover), and mutations inject novelty. There are practical payoffs from this population logic: GAs need only be able to evaluate a candidate (no derivatives), natively accommodate discrete or mixed selections, give hard constraints by means of encodings/repairs/penalties, and extend naturally to multi-objective settings where the result is a Pareto front of non-dominated trade-offs (Deb et al., 2002).

At the same time, gradient/derivative-based methods (steepest descent, Newton, and quasi-Newton; SGD/Adam in deep learning) excel when objectives are differentiable and sufficiently well-behaved. They progress quickly per evaluation, exploit curvature, and under convexity enjoy strong guarantees (Nocedal & Wright, 2006; Bottou, 2010; Kingma & Ba, 2015; Goodfellow, Bengio & Courville, 2016). The practical question, therefore, is not “Which is best?” but “When are gradients the right tool, when are GAs the right tool, and when should we combine them?”.

## Where gradients shine

If the objective is differentiable, constraints are smooth or easily relaxed, and matter speed per evaluation, gradient methods dominate. In large-scale learning, stochastic variants (SGD/Adam) are the engines of modern deep networks (Goodfellow, Bengio & Courville, 2016). With line search or trust regions, they offer stability and (in convex problems) convergence to a global solution (Nocedal & Wright, 2006).

## Where gradients struggle

Many fundamental tasks are not smooth or continuous: selecting a subset of variables; assigning rooms, teachers, or time slots; enforcing prerequisite logic; optimizing rules and policies; balancing competing goals. Black-box simulators and human-in-the-loop evaluations often make derivatives unavailable or unreliable. In rugged landscapes, gradients can stall at local minima or saddle points, and scalarizing multiple goals can hide critical trade-offs.

## Where GAs shine

GAs require only a fitness evaluator. They thrive with non-differentiable, discrete, constrained, or noisy objectives; they are naturally parallel (evaluate an entire population concurrently). This parallelism has been formalized and benchmarked in island-model and master-worker architectures (Alba & Tomassini, 2002; Cantú-Paz, 2001). With algorithms like NSGA-II, they represent multi-objective trade-offs explicitly as a Pareto set (Deb et al., 2002). In practice, this means fewer hand-crafted relaxations, more faithful constraint handling, and better visibility into the trade space.

## Where GAs cost

Evaluation budgets. On smooth, well-behaved problems, GAs often need more evaluations to match gradient quality. Good practice – encoding choices, operator tuning, constraint handling, and principled termination – is therefore part of the method.

## Rule-of-thumb

In summary (see Figure I.1), please note that

- use gradients when the problem is differentiable, and you want fast progress per evaluation.

| | Low discreteness | Mixed | High discreteness | Constraints |
|---|---|---|---|---|
| Differentiability | Gradients (SGD/Adam) | Gradients + restarts | Relaxations + Gradients | Penalized gradients |
| Weak Differentiability | Hybrid (GA + local) | GA / ES | GA / DE / PSO | GA with repairs |
| No Differentiability | GA (binary) | GA (mixed) | Permutation GA | NSGA-II (MO) |

**Figure I.1** - Gradient vs. Evolutionary search domains

Decision matrix showing where gradient-based, evolutionary, or hybrid optimization methods perform best, depending on differentiability, constraints, and discreteness.

- use GAs when the problem is black-box, discrete, constraint-heavy, or genuinely multi-objective, or when decision makers need a diverse set of viable alternatives (not a single point).
- hybridize when the problem mixes both: let GAs explore discrete structures or policies, and use gradients to fine-tune continuous parameters inside each candidate.

## Why GAs still matter in the era of LLMs

Large Language Models (LLMs) do not diminish the role of optimization; they intensify it. LLMs are the result of massive probabilistic gradient-based training (Bottou, 2010; Kingma & Ba, 2015; Goodfellow, Bengio & Courville, 2016) and perform search at inference time, sampling schedules, beam search (Vaswani et al., 2017; Brown et al., 2020; OpenAI, 2023; Holtzman et al., 2019) and tool-augmented inference (Schick et al., 2023; Yao et al., 2022). Around these models, however, we face a thick ring of discrete design choices and constraints: prompt templates, retrieval parameters, cost/latency budgets, fairness and safety thresholds, grading rubrics, and agent routing. This ring is GA territory (see Figure I.2):

- *GAs around LLMs*. Evolve prompts, RAG settings, tool-use policies, rubric thresholds, and latency-quality-cost configurations under explicit constraints.

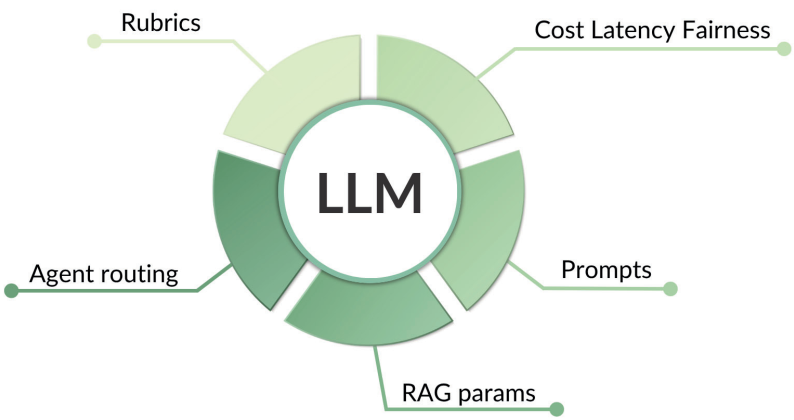


**Figure I.2** - LLM ↔ GA integration architecture

Box-and-arrow schematic linking Large Language Model components with GA outer loops. It illustrates reciprocal roles of LLMs as evaluators or generators inside evolutionary pipelines.

- *LLMs inside GAs*. Treat the LLM as a surrogate evaluator (cheap pre-screening before expensive simulation), a repair operator (turn natural-language rules into feasibility checks and fixes), or a generator of substantial initial populations.
- *hybrid loops*. Alternate GA exploration (discrete structures) with gradient refinement (continuous parameters), borrowing strengths from both paradigms.

## Why Education and Learning Analytics need optimization-as-design

Education abounds with optimization problems: timetabling under hard constraints; curriculum sequencing that respects prerequisites and cognitive load; resource allocation of tutoring or advising; cohort grouping to balance heterogeneity; early-warning systems that must be accurate, parsimonious, and fair; policy optimization that trades learning gains against workload and cost. These are natural GA applications:

- *structure optimization*: feasible timetables and pathways using feasibility-preserving encodings and constraint-aware operators.
- *model optimization*: feature and rule selection that balances predictive power, interpretability, and measurement cost; robust tuning under drift and noise.
- *policy optimization*: multi-objective discovery of non-dominated intervention policies (e.g., learning gains vs. cost vs. equity), visualized as Pareto fronts for transparent governance.

Crucially, GAs promote methodological clarity: they force us to formalize objectives (what we value), encode constraints (what we must respect), and evaluate candidates out-of-sample (what truly generalizes). In Learning Analytics, that triad aligns with accountability, fairness, and reproducibility. For an overview of Learning Analytics practice and challenges see Ferguson, 2012, and Lang et al., 2022; and for effects of Intelligent Tutoring System (ITS) see Ma et al., 2014. See Qu et al., 2009, for a survey specific to examination timetabling.

## About this book

This book is organized around a simple premise: modern AI and data-intensive decision making reduce, again and again, to search and optimization. We must choose among competing possibilities under uncertainty and constraints – models and features, schedules and resources, rules and policies – and we must do so in spaces that are often jagged, partially discrete, and shaped by real-world requirements. The first part of the book lays the theoretical and methodological groundwork for approaching these problems with Evolutionary Computation, and in particular with Genetic Algorithms; the second part translates that groundwork into the educational domain, where learning processes, institutional rules, and ethical imperatives turn optimization from an abstract exercise into a concrete design discipline.

The book offers a practice-oriented synthesis of how Evolutionary Computation, especially Genetic Algorithms, can be used to design and govern complex educational systems under uncertainty. The book advances “optimization-as-design”, using multi-objective methods that surface Pareto fronts so institutions can balance effectiveness, cost, equity, latency, and workload in transparent ways.

The book contains references for a software companion package in R – *geneticaR* – with a R library and several example scripts.

Part I develops the methodological foundations: representation and fitness design; variation and selection; constraint handling via penalties, repairs, and feasibility-preserving encodings; and multi-objective algorithms such as NSGA-II, together with guidance on when to prefer gradient methods, when to wrap them with evolutionary search, and how to scale via parallel and surrogate evaluation.

Part II applies these tools to Learning Analytics and Educational Data Mining, demonstrating curriculum sequencing, timetabling, resource allocation, early-warning and intervention pipelines, collaborative group formation, and intelligent tutoring, often in prescriptive settings where the question is not only “who is at risk?” but “what should we do, when, and under which constraints?”. Throughout, the book treats ethics and governance as first-class design objectives: fairness and privacy are encoded in fitness; documentation, auditability, and human-in-the-loop oversight are emphasized; and recommendations are aligned with emerging regulatory frameworks (e.g., EU AI Act, GDPR).

The text also engages contemporary AI practice by showing how evolutionary search complements LLM-enabled workflows (e.g., prompt and policy tuning with budget and equity constraints), positioning EC not as a competitor to modern learning systems but as the outer-loop that turns values into operational choices. By the end, researchers, data scientists, and institutional leaders will have a coherent toolkit – and realistic guardrails – for building transparent, auditable, and effective optimization pipelines that make educational trade-offs visible and choices defensible.

Part I begins from first principles. It introduces the reader to the idea that optimization can be carried out not only by following derivatives downhill, but also by orchestrating a population of candidate solutions that evolve through selection, recombination, and mutation. The biological analogy is more than a metaphor: it gives us a language for representing solutions (as binary strings, real-valued vectors, or permutations), for generating variability (through crossover and mutation operators suited to each encoding), and for exerting pressure toward improvement (through selection mechanisms that favor fitter candidates without eliminating diversity). By working through these components – representation, initialization, fitness evaluation, selection, crossover, mutation, elitism, and termination – the reader sees how a general, derivative-free search strategy can be adapted to many different problem types, including those where gradients do not exist or are unreliable.

As the exposition deepens, the part situates Genetic Algorithms alongside the broader family of Evolutionary Computation – evolutionary strategies, evolutionary programming, genetic programming, differential evolution – and shows how these paradigms differ primarily in representation and variation operators while sharing the same population-based logic. Practical questions follow naturally: how to tune population size and iteration budgets in light of computational cost; how to diagnose progress and avoid premature convergence; how to handle constraints with penalties, repairs, or feasibility-preserving encodings; and how to make the method reproducible and robust. The discussion culminates in multi-objective optimization. Instead of collapsing multiple goals into a single scalar target, algorithms such as NSGA-II approximate the Pareto front: the set of solutions that cannot be improved on one objective without worsening another. This is not a theoretical detour; it provides a disciplined way to surface trade-offs that genuinely exist in practice – accuracy versus interpretability, performance versus cost, latency versus quality – and to hand decision makers a transparent menu of non-dominated choices.

Throughout Part I, gradient-based methods serve as a foil. Where objectives are differentiable, constraints are smooth, and curvature is informative, gradients are exceptionally efficient and enjoy strong guarantees; they are the engines behind modern large-scale learning. However, many problems that matter – especially those involving discrete decisions, complex rules, or black-box evaluators – do not fit that template. In those settings, the population-based, probabilistic search of Genetic Algorithms is not a poor substitute for calculus; it is the right abstraction. Rather than asking "which approach is best", the text encourages the reader to recognize when each is appropriate, and how to combine them when a problem mixes continuous parameters and discrete structure.

Part II carries this logic into Education and Learning Analytics. Here, optimization is not an optional refinement but a daily necessity. Institutions schedule courses and exams under resource and precedence constraints; instructors and platform designers sequence activities to scaffold knowledge without overwhelming learners; analysts build predictive models that must be accurate yet parsimonious, stable yet fair; administrators allocate scarce support in ways that should be effective, transparent, and equitable. Against this backdrop too, Genetic Algorithms offer three kinds of value listed above: (1) they provide a constructive way to design structures – timetables, groups, curriculum paths – by encoding feasibility directly into candidate solutions and using variation operators that respect institutional rules; (2) they offer a principled approach to model building – selecting features or items, tuning hyperparameters, and reducing dimensionality – with evaluation protocols that emphasize cross-validation and generalization; (3) furthermore, they enable explicit policy optimization – discovering intervention strategies that balance learning gains, cost, and equity, and presenting the results as Pareto fronts rather than as single opaque recommendations.

The applications in Part II unfold methodically. Readers see how to formalize objectives that reflect pedagogical value, not merely clickstream convenience; how to express constraints that mirror real classrooms and platforms; and how to evaluate candidate solutions with the same discipline expected in empirical research. Worked cases cover the spectrum of regression and classification settings – where variable selection and model calibration are central – to latent variable and instrument design – where factor structure and reliability must be balanced against assessment length – to analyses of behavioral tables and cohorts – where correspondence analysis, clustering, and dimensionality reduction serve interpretation and action. Along the way, the narrative remains attentive to reproducibility, fairness, and accountability, treating them as optimization criteria in their own right rather than as afterthoughts.

A consistent thread connects both parts to contemporary AI practice. Large Language Models are trained by massive gradient-based optimization and, at inference time, rely on search procedures to generate and evaluate alternatives. Around them, however, the consequential choices are often combinatorial, constraint-heavy, and multi-objective: how to shape prompts and retrieval, how to set grading and feedback rubrics, how to route agentic workflows within budget and latency limits, how to ensure equitable treatment across subgroups. In precisely these places, the tools developed in Part I and exercised in Part II are directly applicable. Genetic Algorithms can wrap modern models – tuning the discrete and constrained decisions they expose – while modern models can assist evolutionary search by serving as fast evaluators, critics, or repair mechanisms. The result is not a competition between paradigms but a coherent optimization-as-design workflow.

Read in this way, the two parts are not separate silos but two views of a single enterprise. The first equips one with concepts, algorithms, and diagnostic habits for population-based optimization; the second shows how those habits translate into educational decisions that matter to learners, instructors, and institutions. The same ideas – representation, fitness, variation, selection, constraint handling, and multi-objective reasoning – appear in both contexts, scaled and adapted to the specifics at hand. By the end, the reader should not only understand how Genetic Algorithms work, but also recognize when they are the right lens, how they complement gradient methods, and how to build solutions that make trade-offs visible and choices defensible.

# PART I

# AN OVERVIEW ON GENETIC ALGORITHMS AND EVOLUTIONARY COMPUTATION

# Chapter 1

## Introduction

At the core of many different challenges in artificial intelligence (including LLM models), engineering, maths, and data science is the fundamental problem of search and optimization (Mirjalili & Dong, 2020; Vlahavas & Vrakas, 2008; Hou & Luo, 2022).

These two concepts are, at least conceptually, intimately related: searching for a feasible solution is also an optimization problem, where the goal is to find the "best" (or at least acceptably good) candidate from a vast space of possibilities. However, practical problems often have features that make standard, or deterministic techniques – such as gradient descent, exact linear programming, or exhaustive search – computationally prohibitive or theoretically impractical (Bangert, 2012; Burke & Kendall, 2014; Lockett, 2020).

One of the most significant sources of difficulty stems from the exponential or even factorial growth of the space of solutions as the problem size increases. For example, suppose someone tries to list out all the permutations in some routing or scheduling problem (like the traveling salesman problem): it would not take long until the number of arrangements overwhelms even the most advanced supercomputers today. In the realm of continuous optimization, the same "curse of dimensionality" leads computational needs to explode once the decision space reaches the order of thousands, millions, or billions of variables. Moreover, numerous implementations are related to non-fragile, dirty, or highly irregular scenery, in which target forms may be scattered with local optima, and discontinuous or complicated constraints that preclude analytical solutions (Burke & Kendall, 2014; Eiben & Smith, 2015).

Because of above mentioned complexities, heuristic and metaheuristic methods are indispensable. In contrast to strictly mathematically based gradient descent optimization schemes or exhaustive enumeration, these methods generate populations of candidate solutions and iteratively transform them through mutation, crossover, and selection with respect to fitness. This evolutionary search balances exploration (global search) and exploitation (local intensification) (Eiben & Smith, 2015; Whitley, 1994), facilitating escape from local optima and limiting the local search to adapt to evolving or uncertain environments. Combinatorial optimization (e.g., job-shop scheduling, network design) and continuous domains (e.g., control parameter tuning in robotics and hyperparameter optimization in machine learning) are some of the many examples. With the power of these biologically inspired techniques, practitioners can descend the inevitable slope of complexity in modern search and optimization tasks and discover both robust or near-optimal solutions that often outperform exact or gradient-based methods on discrete, non-differentiable, or black-box problems (Lv & Zhao, 2018; Kwakye et al., 2024; Elhaddad & Sallabi, 2011; Chang et al., 2024; Yang, 2010a, 2010b; Zadeh, 1965; Kennedy & Eberhart, 1995; Kirkpatrick, Gelatt & Vecchi, 1983).

Evolutionary computation (EC), a subfield of artificial intelligence, is based on the principles of evolution and adaptation found in nature (Holland, 1975, 1992; Michalewicz, 1996). In this larger research field, Genetic Algorithms (GAs) are the best-known family of algorithms (Eiben & Smith, 2015; Goldberg, 1989; Mitchell, 1995, 1996). They apply variation operators (such as crossover and mutation) and selection mechanisms to populations of candidate solutions in order to iteratively evolve increasingly effective solutions to optimization or search problems. As explained by De Jong (1975) and Goldberg (1989), GAs are based on Darwinian principles of survival of the fittest, which expand on progressive selection based on a fitness competition to create an evolutionary process leading to optimal or near-optimal solutions (see Figure 1.1 for a comparison between biological processes and algorithmic representations).

The formal underpinnings of GAs and EC date back to Holland (Holland, 1975, 1992; Forrest & Mitchell, 2016), who also introduced the notion of "schema" to describe how particular solution patterns survive from one generation to the next. De Jong's (1975) subsequent work examined how different parameter settings affect GA results, setting the stage for guidelines on population sizes, mutation rates, and selection methods. Genetic representation, genetic reproduction, and genetic selection,

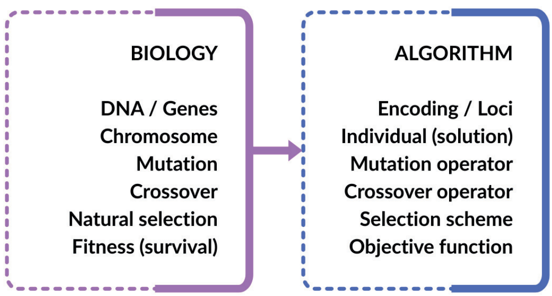


**Figure 1.1** - Biological and algorithmic analogies

Two-column diagram mapping biological evolution concepts (genes, mutation, selection) to algorithmic components (encoding, operators, fitness). Diagram highlights how natural evolution inspires computational design of GAs.

all of which play crucial roles in modern evolutionary algorithms (Eiben & Smith, 2015; Fogel, 1995, 2006), were established by these early contributions.

Other evolutionary methods have been developed over time, and by now, they are referred to as the broader family of evolutionary computation (EC): evolutionary strategies (ES), evolutionary programming (EP), genetic programming (GP), and differential evolution (DE) (Bäck, Fogel & Michalewicz, 1997). All of these paradigms retain the basic idea of iterative search through variation and selection, but vary primarily in how solutions are represented and how variations are applied. These methodologies are still in use today in diverse areas like engineering optimization, operations research, and robotics, as well as artistic activities like the generation of music and art (Goldberg, 1989; Koza, 1992).

In this first part of the book, we will provide a general introductory overview of both GAs and EC by establishing a foundation with the fundamental theories and illustrations from some experiences. By understanding how these biologically inspired principles work, readers can recognize how EC methods can be used to solve complex and often intractable problems using traditional optimization methods.

Chapter 2 addresses evolution on a biological level, outlining the features of natural as well as artificial evolution that make them similar. The third chapter introduces the basic components of a GA: representations, selection mechanisms and genetic operators. We cover other EC paradigms in Chapter 4, such as evolutionary strategies and genetic programming. Chapter 5 discusses implementation considerations like parameter tuning and computational complexity, while Chapter 6 surveys applications in the real world. Advanced subjects, such as multi-objective evolutionary algorithms and coevolution, are covered in Chapter 7, and new research areas and theoretical advances are reviewed in Chapter 8. Finally, Chapter 9 highlights key concepts and potential directions for future exploration of applications in education.

# Chapter 2

# Biological inspiration for Evolutionary Algorithms

Most evolutionary algorithms (EAs), such as Genetic Algorithms, are based on key principles of natural evolution observed in biological systems (Fogel, 1995, 2006; Holland, 1975, 1992). By simulating the processes (mainly inheritance, variation, and selection) that determine how species adapt and flourish across a changing world, researchers have been able to produce highly effective (computational) routines for complex optimization and search (Eiben & Smith, 2015; Goldberg, 1989). This section provides an overview of natural development to illustrate how the ideas behind artificial evolution methods are grounded in biological concepts of genetic coding, mutation, crossover, and selection.

## 2.1 Biological evolution basics

Darwin was the first to articulate in a formal way that natural evolution is powered by two fundamental processes: variation itself, and then that some of those variations actually provide an advantage for survival. In general, people in a population have traits that vary, and some individuals have variant traits that allow them to survive or reproduce more effectively, thus passing their genes onto the next generation (Darwin, 1859). Over generations, this can lead to complex adaptations that help populations better fill their ecological niches.

From a computational perspective, each possible solution to a problem is considered as an "individual" in the population. The implementation of the algorithm facilitates a form of mutation and recombination that introduces variation and allows only the fittest individuals to mate. Thus, the algorithm can *pass* to better solutions even though it converges (Eiben & Smith, 2015; Fogel, 1995, 2006). The current methodology is particularly valuable in situations with a large or complex search space that resists fully enumerating all possible solutions, and in situations where we may not have complete knowledge concerning the optimal function (Goldberg, 1989; Michalewicz, 1996).

## 2.2 Genetic representations in nature

Genetic information in biological systems is encoded in DNA (or RNA in some viruses) as long nucleotide chains – sometimes billions of bases – packaged into chromosomes; the genes they contain regulate protein synthesis and, ultimately, phenotype (Ridley, 2006). Heritable variation (Darwin, 1859) together with population-level genetic variation (Ridley, 2006) provides the raw material on which natural selection operates, a framing echoed in computational treatments (Michalewicz, 1996).

Translating these ideas into algorithmic terms, evolutionary approaches represent candidate solutions as "chromosomes" or "genomes". In Genetic Algorithms, these are often binary strings (or bitstrings), though real-valued vectors and other specialized encodings are also common when the problem demands them (Goldberg, 1989; Holland, 1975, 1992). The encoding should support genetic operators (e.g., crossover and mutation) and provide a clear genotype-to-phenotype mapping back to the problem domain (Eiben & Smith, 2015).

## 2.3 Mutation, Crossover, and Natural Selection

### Mutation

Mutation occurs in nature as a result of evolutionary processes, environmental influences, and various biological processes (Ridley, 2006). Although many mutations will not alter the phenotype and are at least expected to be deleterious, it is also possible that new advantageous mutations might have occurred contributing to evolutionary change. In Genetic Algorithms, mutation is again introduced as small random changes to the representation of the solution. The mutation operator introduces random modifications to the solution configuration (for instance, flipping a bit in a binary string or mutating a real-valued element) to form a new, potentially better solution (Goldberg, 1989; Holland, 1975, 1992). Mutation maintains the population's diversity, helping to avoid premature convergence on sub-optimal solutions (Eiben & Smith, 2015).

### Crossover (Recombination)

In nature, during reproduction, recombination or crossover occurs, in which the genetic material from two parents mixes, and offspring inherit traits from both parents (Ridley, 2006). In Genetic Algorithms (GAs), this is mimicked by choosing two "parent" solutions and swapping sections of their genetic representations (say, bitstrings). The operator takes advantage of existing structures found in high-quality solutions, which may accelerate convergence to a good or optimal design (Goldberg, 1989; Michalewicz, 1996).

### Natural Selection

People who are better suited to their environment are more likely to survive and reproduce. Selection is modeled by favoring solutions with more excellent "fitness" or performance relative to a predefined objective function (Fogel, 2006; Holland, 1975, 1992). Stochastic procedures, including roulette wheel selection, tournament selection, rank selection, are used to probabilistically pick which individuals will pass their genetic code to the next generation (Eiben & Smith, 2015).

These three biological concepts – mutation, crossover, and selection – form the basis of evolutionary algorithms that provide strong performance across runtime pressures and in complex search spaces. By trading off between the bad solution traits that need to be deleted (exploitation) and the introduction of new solution traits (exploration), EAs are able to refine candidate solutions adaptively, in a manner similar to natural evolution (Goldberg, 1989; Michalewicz, 1996).

# Chapter 3

## Genetic Algorithms: fundamentals

One of the parent evolutionary techniques is Genetic Algorithm (GA), which makes use of biologically inspired search mechanisms to solve computational problems (Goldberg, 1989; Holland, 1975, 1992). While a GA uses a population of candidate solutions, it calls on the genetic operators such as crossover and mutation, invoked with a selection mechanism to evolve the initial population towards higher fitness (Eiben & Smith, 2015). This subsection describes the various components of GAs, specifically representation and initialization, fitness evaluation, selection schemes, crossover operators, mutation operations, elitism, and termination criteria (a schematic flow diagram for a GA is shown in Figure 3.1).

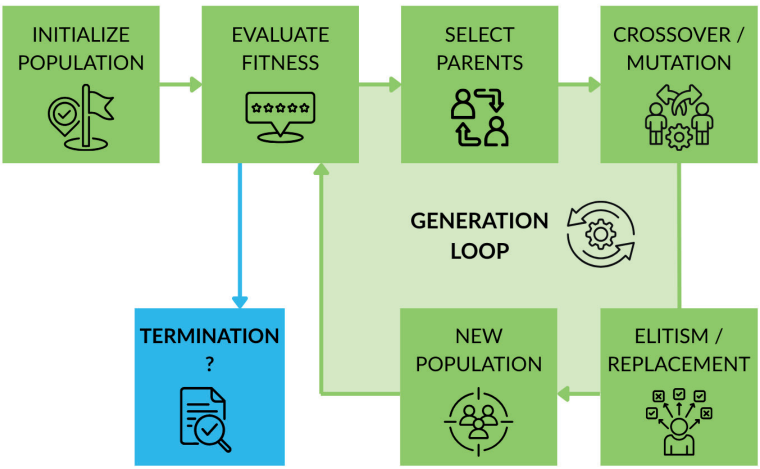


**Figure 3.1** - Genetic Algorithm Workflow

Circular flow illustrating main GA phases: initialization, evaluation, selection, crossover, mutation, and termination. It summarizes the iterative population-based optimization cycle.

## 3.1 Representation (encoding of solutions)

The first and most important step is getting the proper representation for a Genetic Algorithm (GA), which encodes how a potential solution is represented as data to be processed by an algorithm. A common encoding mechanism, which is most often present in the literature, is the classical bitstrings format, where each bit (either 0 or 1) represents a state of some parameter or characteristic (Holland, 1975, 1992). One of the key advantages of the binary encoding is its simplicity, which allows for using directly genetic operators such as single-point crossover and bit-flip mutation (Goldberg, 1989).

However, other representation techniques are also in use to encode it. Real-coded Genetic Algorithms, which use floating-point vectors to represent continuous parameters, are well suited for numerical optimization problems (Michalewicz, 1996). For combinatorial challenging problems like the traveling salesman problem or task scheduling, permutation/sequence encodings (e.g., random keys, order-based) offer a good solution (Gen & Cheng, 1997; Eiben & Smith, 2015). The chosen encoding method can affect which genetic operators are effective and how well the algorithm is able to traverse the solution space (Mitchell, 1996). Table 3.1 compares different encodings with respect to concrete applications in education.

## 3.2 Initialization (population setup)

Having settled on an encoding, the GA starts by creating a random initial set of candidate solutions (the population). The simplest way of initializing a population (Goldberg, 1989) is to randomly initialize each representation individually. This implies a broad search space coverage since the beginning, leading to diversity.

In many applications, a heuristic-based or partially seeded initialization is more successful (Michalewicz, 1996). If, for example, domain knowledge is available a priori to the optimization task, then embedding known reasonable solutions into the initial population can be beneficial due to the so-called seeding in Genetic Algorithms (Eiben & Smith, 2015). Ultimately, whether using group variation or network variation, the aim in populating is to achieve enough variety to prevent premature convergence.

## 3.3 Fitness evaluation

Fitness evaluation is the determination of how well a candidate solution meets the problem-specific objectives (Goldberg, 1989). The efficiency and effectiveness of Genetic Algorithms (GAs) depend considerably on the objective function used for evaluation.

**Table 3.1** - Solution encodings for Genetic Algorithms in educational optimization
Comparison of binary, real-valued, permutation, and mixed encodings, with guidance on when each representation is appropriate, concrete education-focused examples (e.g., item/test selection, timetabling, curriculum sequencing), and notes about feasibility preservation and operator compatibility.

| Encoding | Typical problem | Education example | Notes |
|---|---|---|---|
| Binary | Discrete, yes/no choices; subset selection; Boolean rules | Test-item selection under a blueprint; feature selection for early-warning models; on/off intervention policies | Good for feature flags; easy penalties/repairs; Hamming neighborhoods are natural for Boolean logic |
| Real-valued | Continuous parameter tuning | Cut-scores and grade thresholds; weighting rubrics; time/effort budgets per activity | Bounds and scaling matter; combine with SBX/BLX-α crossover and Gaussian mutation; project to bounds |
| Permutation | Ordering/ sequencing without repetition | Exam timetabling; course sequencing; question orderings in adaptive quizzes | Use order-preserving crossover (PMX, OX, CX) and swap/insert/inversion mutations to preserve feasibility |
| Mixed | Hybrid (some discrete, some continuous) | Curriculum path design (choose modules + continuous workload targets); resource allocation with thresholds | Combine sub-encodings; use problem-aware crossover/repair; often paired withMOEA for trade-offs |

In some scheduling problems, fitness can be a number that represents the total time or other cost required to finish a set of tasks (Michalewicz, 1996). In relation to designing a neural network, fitness could be described as classification accuracy or error metrics of testing in the validation dataset (Eiben & Smith, 2015). Since GAs typically require a large number of evaluations, the computational cost is substantial, especially if the fitness function involves heavy simulations or complex data processing (Mitchell, 1996).

Thus, the fitness function being optimized or approximated is a practical issue for most real-world GA applications. For costly fitness functions, surrogate-assisted EAs can reduce complexity by approximating the true fitness value (Jin, 2011).

## 3.4 Selection mechanisms (Roulette Wheel, Tournament, Rank)

Selection is the process by which some individuals within the extant population are selected to pass their genetic material to the next generation, driving evolutionary change over generations. Various methods are employed:

- *Roulette Wheel (Proportionate) Selection*: select an individual with a probability of selection proportioned to fitness relative to the total population fitness value (Goldberg, 1989). This approach is conceptually simple but can sometimes suffer from scaling issues and premature takeover when few individuals dominate fitness (Eiben & Smith, 2015).
- *Random Tournament Selection*: a random subset (tournament group) is selected from the population, and an individual with the highest fitness value in this group is chosen for mating. It aims at keeping diversity, while putting various pressures on all species (Eiben & Smith, 2015).
- *Rank Selection*: fitness amounts are utilized to assign individuals, and individual probabilities select the individuals according to their relative ranks instead of their absolute fitness value (Bäck, Fogel & Michalewicz, 1997). This approach provides a solution to the scale discrepancy problem of the fitness function.

Selection methods (a comparison is shown in Table 3.2) have to balance between the exploration of different parts of the solution space and using subjective knowledge,

**Table 3.2** - Selection mechanisms: pressure, cost, and robust defaults

Side-by-side overview of roulette (proportionate), tournament (k=2-5), and rank-based selection, indicating relative selection pressure, computational cost, practical advantages and drawbacks, and safe default settings.

| Method | Selection pressure | Cost | Pros/Cons | Defaults |
|---|---|---|---|---|
| Roulette (proportionate) | Low-moderate; depends on fitness scaling | O(N) after normalization | Simple; sensitive to outliers and scaling; can lose diversity if fitness is highly skewed | Apply fitness scaling (sigma or rank) before use |
| Tournament (k = 2–5) | Tunable via k (↑k ⇒ ↑pressure) | O(N·k); very cheap and parallel friendly | Insensitive to absolute scaling; easy to implement; too-high k accelerates convergence | k = 3-4; with-replacem ent tournaments |
| Rank (linear or exponential) | Low-moderate; controlled by rank function | O(N log N) (sorting step) | Robust to scaling; prevents domination by few elites; may slow progress on easy problems | Linear rank; selective pressure ≈ 1.5–2.0 |

selecting for the best individuals only (Whitley, 1994; Mitchell, 1996) in order to avoid one extreme (and get too much randomness) or another extreme (end up with looking at a few elite specimens).

## 3.5 Crossover (Single-Point, Multi-Point, Uniform)

Crossover or recombination combines genetic information from two parent solutions to produce progeny. Combining the desired traits of different parents allows for speeding up the convergence and spotting high-quality solutions earlier (Goldberg, 1989).

Common types of crossover include:

- *Single-Point Crossover*: split at one cut-point; offspring take the left prefix from one parent and the right suffix from the other (and vice versa) (Holland, 1975, 1992).
- *Multi-Point Crossover*: k-point crossover alternates segments between parents, increasing disruption relative to 1-point (Michalewicz, 1996).
- *Uniform Crossover*: for each gene of the offspring, it chooses one of the corresponding genes from the parents with equal probability (Eiben & Smith, 2015).

The choice of crossover operator (see Table 3.3) can affect the trade-off between exploiting already found "building blocks" and searching for new genetic diversity (Goldberg, 1989).

## 3.6 Mutation (Bit-Flip, Swap, and others)

Mutation produces small random changes in organisms, which increase the diversity on the population level (Holland, 1975, 1992). With this feature, the Genetic Algorithm (GA) sustains diversity and helps escape local optima by injecting stochastic variation (Mitchell, 1996). Examples of mutation include (see Table 3.4 for a summary):

- *Bit-Flip Mutation*: the replacement of a zero with a one or vice versa in binary encoding (Goldberg, 1989);
- *Swap Mutation*: two selected elements of a permutation are swapped, for example, in routing or scheduling problems (Michalewicz, 1996);
- *Gaussian Mutation (for real-valued encodings)*: one or more genes each receive a random value drawn from a Gaussian (Eiben & Smith, 2015).

Mutation rates are generally problem- and encoding-dependent and best estimated by recourse to experimental methods. A mutation rate that is too high can result in disruption of beneficial structures, whereas a rate that is too low may cause premature convergence (Bäck, Fogel & Michalewicz, 1997).

**Table 3.3** - Crossover operators across representations and their feasibility properties

Matrix of crossover operators – 1/2-point and uniform (binary), SBX and BLX-α (real-valued), and PMX/OX/CX (permutation) – showing where each operator applies and whether it preserves feasibility. Abbreviations: SBX = Simulated Binary Crossover; dX-α = Blend Crossover with parameter α; PMX = Partially Mapped Crossover; OX = Order Crossover; CX = Cycle Crossover; TSP = Traveling Salesman Problem.

| Operator | Binary | Real | Permutation | Preserves feasibility | Notes |
|---|---|---|---|---|---|
| 1/2-point | ✔ | | | No | Exploits "building blocks" (schema); standard for bitstrings |
| Uniform | ✔ | | | No | Maximizes mixing/diversity; can disrupt useful linkages |
| SBX | | ✔ | | Yes* | Simulated Binary Crossover *with bound handling/clipping |
| BLX-α | | ✔ | | Yes* | Blend crossover; tune α *respect bounds via projection/reflection |
| PMX | | | ✔ | Yes | Partially mapped crossover; preserves absolute positions |
| OX / CX | | | ✔ | Yes | Order/cycle crossover; preserves relative order; excellent for TSP/timetabling |

**Table 3.4** - Mutation operators, calibration, and diagnostics for diversity control

Mapping from encoding to mutation choice-bit-flip (binary), Gaussian (real), swap/insert/inversion (permutation) including typical rates or step sizes, their impact on population diversity, and practical diagnostics for increasing/decreasing mutation when convergence stalls or schemas are disrupted. Notation: L denotes chromosome length.

| Operator | Encoding | Typical rate | Effect on diversity | When to adjust |
|---|---|---|---|---|
| Bit-flip | Binary | Per-bit $p \approx 1/L$ ($L$ = chromosome length) | Maintains exploratory variability | ↑ if premature convergence; ↓ if schemas repeatedly disrupted |
| Gaussian (additive) | Real | Per-gene mut. prob. 0.1-0.3; $\sigma \approx$ 5–20% of range | Local refinement with controllable step size | Decrease σ as population converges; increase when stagnation is detected |
| Swap / Insert / Inversion | Permutation | 1-3 mutation operations per offspring ($p \approx$ 0.1-0.3) | Preserves feasibility; reshapes tours/orders | Increase when diversity drops; reduce if good partial orders are destroyed |

## 3.7 Elitism and replacement schemes

We speak of elitism when top performers of a single generation, as an elite legacy (Goldberg, 1989), are passed over to the new generation without change (typically 1-5%). GAs having retained elite solutions are able to maintain progress already achieved, so constraining the possibility of losing high-fitness individuals due to random fluctuations in selection or genetic operations (Eiben & Smith, 2015).

In replacement schemes, new offspring take the place of current individuals. We observe a generational replacement when the entire population at a time endows the next generation. In a steady-state GA, only a few individuals are replaced at any time, which makes the population significantly overlapping between generations (Bäck, Fogel & Michalewicz, 1997). Every scheme influences the balance of exploration and exploitation.

## 3.8 Termination criteria

GAs are iterative until a termination condition is met (Michalewicz, 1996).

Common criteria include:

- *fixed number of generations*: the algorithm stops running after a given number of generations (Goldberg, 1989).
- *end fitness*: as long as we fulfill the condition that has the best fitness in our population, we can complete the process (Mitchell, 1996).
- *convergence measures*: the search ends if no, or only marginal progress is observed in fitness over multiple generations (Eiben & Smith, 2015).
- *limitations of time or resources*: computation time or resources availability may also place a practical hard limit on the lifespan of the run (Fogel, 1995, 2006).

The choice of termination criterion, therefore, usually embodies a compromise between expenditure on the foregoing computational costs (Bäck, Fogel & Michalewicz, 1997).

# Chapter 4

# Broader Evolutionary Computation paradigms

Though not without competition, Genetic Algorithms (GAs) are among the most widely used Evolutionary Computation (EC) methodologies. All these approaches follow the underlying principle of iterative evolution through variation and selection, differing in representation schemes, variation operators, and domains of applicability (Bäck, Fogel & Michalewicz, 1997; Eiben & Smith, 2015). This section will show some of the major paradigms and include what is excellent about them, along with situations where they may shine the most. Table 4.1 summarizes and compares the main features of the paradigms described here.

## 4.1 Genetic Programming (GP)

Genetic Programming (GP) is an evolutionary algorithm that expands evolutionary search to the space of programs/expressions rather than fixed-length parameter vectors (Koza, 1992).

This improvement allows GP to address a variety of symbolic regression, classification, and automated algorithm and controller design tasks.

### Tree-based representations

In GP, solutions are usually represented as syntax trees whose internal nodes indicate functions (e.g., arithmetic or logical operators) and whose leaf nodes represent inputs or constants (Koza, 1992). Evolution can grow them or prune them off, creating programs of different degrees of complexity that increase over time. Crossovers are done by exchanging subtrees from parent individuals, whilst mutations might be replacing a randomly selected subtree with a newly generated subtree (Banzhaf et al., 1998).

### Evolving programs vs. Parameters

Unlike GAs that refine a fixed-size chromosome of parameters, GP evolves the structural composition of programs – their shape – and is thereby capable of flexible adap-

**Table 4.1** - Comparison among evolutionary computation paradigms

Genetic Algorithms (GA), Genetic Programming (GP), Evolutionary Strategies (ES), Evolutionary Programming (EP), Differential Evolution (DE), and Memetic Algorithms (MAs) are differed based on representations/encodings, operators (mutation, crossover), applications and advantages.

| | Representation | Key Operators | Applications | Advantages |
|---|---|---|---|---|
| Genetic Algorithms (GA) | Bit strings or real-valued vectors | Crossover, mutation | Combinatorial/ general optimization | Robustness |
| Genetic Programming (GP) | Tree-based programs | Crossover, mutation | Symbolic regression, algorithm creation | Creativity |
| Evolutionary Strategies (ES) | Real-valued vectors | Mutation | Parameter optimization | Robustness |
| Evolutionary Programming (EP) | Real-valued vectors | Mutation | Function optimization | Simplicity |
| Differential Evolution (DE) | Real-valued vectors | Differences | Continuous or multimodal optimization | Simplicity Efficiency |
| Memetic Algorithms (MAs) | Any (combined with local search) | Crossover, mutation, local optimizer | Complex problems/Hybrid search | Solution quality |

tation in both structure and content (Koza, 1992). This is generally achieved at the cost of increased computational overhead, larger trees are more expensive to score. Nevertheless, the ability of GP to evolve new topologies has led it to be a powerful tool in fields where the solution space is not just tuning parameter values but also finding an appropriate functional form (Tran, Xue & Zhang, 2019; Eiben & Smith, 2015).

## 4.2 Evolutionary Strategies (ES)

Evolutionary Strategies (ES) were introduced in the 60s and 70s (mainly in Europe) as a strategy for optimizing parameters with real values (Bäck, Fogel & Michalewicz, 1997; Rechenberg, 1973). Specifically, it tailors the strategy parameters that influence the mutation rates, making this optimization technique well-suited for continuous tasks encountered in engineering and applied sciences (Schwefel, 1995).

### Real-valued representations

ES more commonly works with real-valued vectors, unlike classical GAs, which frequently use binary encodings. Each individual is made up of decision variables (the components of the solution) and their strategy parameters, such as standard deviations for Gaussian mutations (Beyer & Schwefel, 2002). This direct encoding is often more suitable for continuous parameter spaces (Eiben & Smith, 2015).

### Self-adaptation of strategy parameters

ES adopts a self-adaptation principle: the algorithm can vary the intensity of exploration within each solution's neighborhood by evolving its strategy parameters (e.g., mutation step sizes) (Bäck, 1996). On real-valued problems, the self-adaptive mechanism provides a faster and more robust convergence in many cases. A fixed-parameter GA often achieves faster and more robust convergence on real-valued problems than naïvely tuned fixed-parameter GAs (Beyer & Schwefel, 2002).

## 4.3 Evolutionary Programming (EP)

Lawrence J. Fogel (1966; 1995) first developed Evolutionary Programming (EP) as a tool for evolving finite state machines, subsequently extended to numeric optimization and other problems. Mutation used to be the leading variety operator in EP, with crossover often absent. EP can be generalized to different types of representations, such as real-valued encodings for individuals. Like ES, EP uses self-adaptive mutation parameters to balance exploration over exploitation (Fogel, 1995, 2006). EP has been used in control, game strategy, and data classification (Bäck, Fogel & Michalewicz, 1997).

## 4.4 Differential Evolution (DE)

Another real-valued optimization algorithm is Differential Evolution (DE), that has been presented by Storn & Price (1997). It maintains a set of parameter vectors to which variation is added by subtracting differences of randomly picked population members and then adding them to a base vector. Only when the offspring is fitter do they replace their parent (Price, Storn & Lampinen, 2005). It is acknowledged for its simplicity (requiring fewer parameters than most other evolutionary algorithms) and its strong performance in the face of continuous or multimodal optimization problems (Eiben & Smith, 2015).

## 4.5 Memetic Algorithms (MAs): hybrid EAs with local search

Moscato (1989) introduced the concept of Memetic Algorithms (MAs), which combine global evolutionary search with local improvement heuristics.

They are sometimes called a "hybrid genetic algorithm" or simply "genetic local search". The idea is that intermittent local search steps can be used to improve the quality of promising solutions and make them searchable by global exploration through evolutionary operators (Hart, Smith & Krasnogor, 2005). Such hybridization techniques have been found to be quite efficient for problems, including complex combinatorial problems like the traveling salesman problem and scheduling, frequently outperforming baseline EAs on many hard combinatorial problems (e.g., Merz & Freisleben, 1999).

# Chapter 5

# Implementation considerations

To design a good evolutionary algorithm (EA), you need more than choosing among genetic operators. In addition, practitioners need to set hyperparameters, handle the complexity of computations, and select appropriate tools or libraries to support development (Eiben & Smith, 2015; Michalewicz, 1996). This section deals with how to handle these practical issues, find trade-off between exploration and exploitation, manage runtime costs, and exploit existing software frameworks.

## 5.1 Parameter tuning (population size, mutation rate and more)

Parameter tuning is one of the most critical aspects (and often, most time-intensive) of evolutionary computation. Population size, mutation rate, crossover probability, and selection pressure are some of the main choices that have a dramatic impact on the performance of the Genetic Algorithm (De Jong, 1975; Goldberg, 1989):

- *size of population*, the number of candidate solutions at each generation: a larger population usually provides more genetic variability, therefore reducing the possibility of premature convergence. However, this also raises the computational cost per generation. Consequently, the diversity vs average speed trade-off has to be found (Mitchell, 1996; Eiben & Smith, 2015).
- *mutation rate,* the probability that a gene in a chromosome is randomly altered during mutation: if the rate of mutation is too high, it can disrupt the inheritance of beneficial traits and become like a random search. A mutation rate that is too low can instead cause stagnation as the population converges to homogeneity (Bäck, Fogel & Michalewicz, 1997).
- *crossover probability*, the probability that two parent chromosomes are recombined to produce offspring: in classic binary GA practice this is usually set in the range of 0.6-0.9 (Goldberg, 1989). It has to be fine tuned for real-coded/permutation encoding and for certain domains might be better served through adaptive, or problem-specific crossover strategies (Eiben & Smith, 2015).

- *selection pressure*, the degree to which individuals with higher fitness are favored during the selection process: the methods of selection, like tournament or rank-based selection, can be twisted to have high or low pressure. High-pressure selection will quickly identify the elite individuals, but may limit diversity; low-pressure selection could broaden the search while preserving greater variability (Mitchell, 1996; Eiben & Smith, 2015).

These parameters are closely interconnected. To determine the appropriate parameter settings, it is necessary to use meta-optimization (i.e., using one EA to tune another), experimental testing, or domain-specific heuristics (Michalewicz, 1996). This tuning process can be simplified using automated tools and statistical methods, such as Design of Experiments (DOE) or Bayesian optimization (Snoek, Larochelle & Adams, 2012; Eiben & Smith, 2015). In Table 5.1, some guidelines for parameter optimization are reported (Snoek, Larochelle & Adams, 2012; Eiben & Smith, 2015).

**Table 5.1** - Resource-aware parameters of evolutionary runs under evaluation budgets

Guidelines for choosing population size, generation count, and parallelization strategy across four fitness-cost regimes (low → very high), together with recommendations on surrogate modeling and caching.

| Fitness cost regim | Population | Generations | Parallelization | Surrogates Caching | Notes |
|---|---|---|---|---|---|
| Low | 80–120 | 200–500 | Optional | Not needed / lightweight caching | Fast iteration; emphasize exploration and operator ablations |
| Moderate | 120–200 | 300–800 | Recommended (threads/GPU) | Useful (memoization; occasional surrogate) | Balance pop×iters; monitor diversity to avoid premature convergence |
| High | 200–400 | 100–300 | Essential (cluster/ archipelago) | Critical (Kriging/RF; reuse evals) | Favor larger pops, fewer generations; asynchronous islands help |
| Very high | 160–240 | 400–800 | Essential (asynchronous) | Strongly advised (active learning + cache) | Exploit surrogate to pre-screen; report wall-clock + eval counts |

## 5.2 Computational complexity

Evolutionary algorithms have a higher computational cost, particularly for large-scale or high-dimensional tasks (Bäck, Fogel & Michalewicz, 1997). At every generation, the algorithm assesses the fitness of all population members and then applies selection and variation operators to create the next cohort. With a large population and an expensive fitness function, the cost per generation scales directly with both the number of individuals and the average evaluation time per individual (Mitchell, 1996). This makes the total time for the whole run also scale with the number of generations executed (Deb et al., 2002; Zitzler, Laumanns & Thiele, 2001).

In many practical applications, the fitness evaluation is a dominant cost over the genetic operators themselves. If an objective requires either a simulation, a numerical solver, a cross-validation loop, or data-intensive preprocesses, then each individual may take quite some time to assess. On the other hand, when the objective is a closed-form expression that can be computed very quickly, the costs of selection, crossover, and mutation may become more noticeable, although still minuscule compared to evaluation. In multi-objective variants, overhead grows faster than linearly with population size due to dominance ranking and diversity maintenance, but is typically less significant compared to simulation costs.

This could be exacerbated or mitigated by a number of factors. Some constraint-handling methods that repair or project infeasible candidates can incur additional solver calls – for example, resampling schemes (e.g., repeated cross-validation) used in feature optimization. Model selection further increases the number of fitness assessments per individual that are required. For this reason, parallel and distributed execution (across threads, GPU, or clusters) is your friend as it enables you to evaluate many individuals at the same time. It uses island models to help combat synchronization bottlenecks and facilitate load-balanced evaluations when evaluation times vary significantly. Other surrogate-assisted methods can model the fitness landscape using statistical models to make expensive evaluations by focusing them on promising regions of the search space (Jin, 2011). Caching and memoization help when similar or near-duplicate candidates recur; vectorization and batched evaluation reduce the interpreter overhead in high-level environments (Alba & Tomassini, 2002; Cantú-Paz, 2001; Biscani & Izzo, 2020). Table 5.2 reports a list of typical troubleshooting and suggestions on causes and how to fix.

Moreover, finally, practical complexity is more than just plain time. In addition, large populations, long representations, and the regular logging/checkpointing required to ensure reproducibility (fixed seeds) can add significant I/O costs; computational resources also become heterogeneous, limiting throughput when reproducibility constraints (fixed seeds, deterministic reductions) are imposed. As a result, more and more careful papers highlight how many fitness evaluations it uses and how much wall-clock time it needs to reach a certain desired level, apart from variables that affect the performance: population size, number of generations, and details of

**Table 5.2** - Troubleshooting evolutionary runs and reporting essentials for reproducibility

Diagnostic mapping from common symptoms (premature convergence, fitness stalls, overfitting, infeasibility, unstable Pareto fronts, excessive runtime) to likely causes and concrete remedies, plus a checklist of what to report (diversity metrics, restart policy, validation protocol, Pareto spread, hardware/parallelism, evaluation counts). Abbreviations: CV = cross-validation.

| Symptom | Likely cause | Fix | What to report |
|---|---|---|---|
| Premature convergence | Excessive elitism; low mutation/diversity | Lower elitism; raise mutation; diversify seeding | Fitness-over-time; diversity metrics (entropy/Hamming); elite retention policy |
| Fitness stalls | Insufficient exploration; deceptive landscape | Restart with new seeds; island model; operator mix | Operator settings; restart policy; islands/migration schedule |
| Overfitting | Objective mis-specification; weak validation | Use CV/repeated CV; add parsimony objective/penalty | CV protocol and variance; selected model size; test risk |
| Infeasible population | Weak constraint handling | Stronger repair; feasibility-preserving encoding | Feasible-rate per generation; violation magnitudes; $\lambda$ if penalized |
| Unstable Pareto front | Too few evaluations; noisy fitness | Increase evals; smoothing/ bootstrapped fitness | Pareto spread/spacing; knee point selection criteria; number of evals |
| Excessive runtime | Expensive fitness; serial evaluation | Parallel/islands; surrogates; caching memoization | Wall-clock and eval counts; hardware; parallelism strategy |

implementation (Bäck, Fogel & Michalewicz, 1997; Mitchell, 1996; Luke, 2013). For instance, in various fields such as complex simulations, computational fluid dynamics, or large-scale data analysis, fitness evaluations themselves can be the bottleneck (Eiben & Smith, 2015).

There are many strategies to deal with high computational costs:

- *parallelization*: many individual evaluations can be performed at the same time (Fogel, 1995, 2006; Alba & Tomassini, 2002; Cantú-Paz, 2001; Biscani & Izzo, 2020). With modern frameworks that support execution in a distributed or cloud-based manner, the evolution can be even faster.
- *surrogate model*: approximate models, such as neural networks or regression trees, can serve as a surrogate to replace a costly fitness function, thereby reducing the number of exact evaluations needed (Jin, 2011).

- *hybridisation with local search*: memetic algorithms can use the local search method for more rapid convergence of promising individuals and then potentially have to run for fewer generations overall (Moscato, 1989).

Through practice, we found that if the controlling parameters, such as population size, generations, and operator settings, are correctly balanced, then it allows the total lifetime of the program to be controlled without sacrificing output quality at a central level (Eiben & Smith, 2015).

## 5.3 Coding and frameworks (pre-existing toolkits, libraries)

In reality, developers (at least in academia or industry) implement evolutionary algorithms using libraries and frameworks that provide data structures, genetic operators, and utilities for parameter tuning (Luke, 2013). This method aims to minimize development costs and result in trustworthy, tested implementations (Eiben & Smith, 2015) as libraries and Toolkits in Java, Python, R, MATLAB. We describe some solutions in the next paragraphs and summarize them in Table 5.3. A detailed overview of KEEL - Knowledge Extraction based on Evolutionary Learning (Alcalá-Fdez et al., 2009, 2011), a Java open source software for evolutionary approaches and soft computing, is provided in Chapter 14.

### ECJ (Evolutionary Computation in Java)

A battle-tested, flexible library that comes with an extensive set of algorithmic primitives (e.g., selection, crossover, mutation) as well as native support for executing algorithms in parallel (Luke, 2013).

### DEAP (Distributed Evolutionary Algorithms in Python)

A Python library meant for fast prototyping of evolutionary algorithms featuring built-in genetic operators, parallelization support, and a flexible architecture (Fortin et al., 2012).

### R Packages

- *geneticaR*: a companion R package for this book (see Appendix) that proposes single and multi-objective Genetic Algorithms (including NSGA-II), supports constrained and unconstrained problems with binary/real/permutation encodings, offers penalty- and repair-based constraint handling, and includes plotting, reproducibility utilities, and Learning-Analytics–oriented examples.
- *GA*: a comprehensive R package implementing Genetic Algorithms for constrained and unconstrained optimization problems and based on various selection, crossover, or mutation concepts (Scrucca, 2013).

**Table 5.3** - GA/EC software toolkits and capabilities relevant to practice

Concise inventory of widely used libraries (geneticaR, GA, rgenoud, DEAP, ECJ, and pygmo), indicating language, objective support (single/multi-objective), constraint handling, parallel features, license, and distinctive notes.

| Library | Language | Single/Multi Objective | Constraints | License | Notes |
|---|---|---|---|---|---|
| geneticaR | R | Both (incl. NSGA-II) | Penalty / repair | Open Source | Book companion; LA-oriented examples; reproducibility tools |
| GA | R | Both | Penalty (bounds/ penal) | GPL-2 | Strong baseline for many tasks |
| rgenoud | R | Both (GA; gamultiobj-style workflows) | Penalty; hybrid local | GPL-2 | Combines GA with derivatives where available |
| DEAP | Python | Both (plug-in NSGA-II, etc.) | Custom (user-defined) | BSD | Rapid prototyping; rich ecosystem integration |
| ECJ | Java | Both | Custom (user-defined) | GPL | Mature, flexible; extensive primitives/ parallellism |
| pygmo | C++ / Python | Both (many global optimizers; MOEAs) | Custom (problem wrappers) | LGPL | Archipelago model; asynchronous island evolution |

- *rgenoud:* an R package that combines genetic optimization with derivative-based methods, where applicable; practical for statistical estimation and other optimization problems (Mebane & Sekhon, 2011).

## MATLAB Toolboxes

Global Optimization Toolbox comes with a built-in genetic algorithm solver (GA) and multi-objective GA solver (gamultiobj), as well as other global optimization methods (MathWorks, 2025). The hybrid algorithm is provided, and users can customize selection schemes, as well as crossover and mutation functions. They can hybridize the optimization strategy with local search methods (e.g., pattern search) from the easy-to-use MATLAB interface.

### pagmo/pygmo (C++ core with Python bindings)

Provides a unified interface to a large collection of global optimizers (evolutionary, swarm, and other metaheuristics) and ready-made benchmark problems, and supports parallel/asynchronous execution via an archipelago (island) model (Biscani & Izzo, 2020; Izzo, 2012).

### GAlib (C++ Genetic Algorithms Library)

GAlib is a long-standing, open-source C++ toolkit for building genetic-algorithm–based solvers. It provides a compact set of abstractions for genomes (solution encodings), genetic operators (selection, crossover, mutation), and algorithm shells (e.g., generational and steady-state variants), so that users can assemble problem-specific optimizers without re-implementing the GA plumbing. At a high level, you (i) select or define a genome type suitable for your decision variables, (ii) specify an objective/fitness function, (iii) choose the GA variant and operators; then (iv) execute the search with tunable parameters including population size, crossover/mutation rates, and termination. This design makes GAlib suitable for classroom instruction, prototyping, and embedding standard GA pipelines into a C++ system.

### In-house or custom implementations

Many projects also custom-develop EA code, especially when domain-specific representations and/or performance optimizations are required. This is the most flexible solution, but also usually involves more work by the developers since this policy contains complementary overhead costs (Bäck, Fogel & Michalewicz, 1997).

The choice of a framework or library is influenced by factors such as the programming language in which they are written, ease of use, community support for development, and computational efficiency (Mitchell, 1996). In many cases, working from an existing toolkit offers a solid foundation for practitioners to build upon when it comes to application-specific improvements.

# Chapter 6

# Practical applications

Evolutionary algorithms (EAs) are popular across many disciplines and are well applied to the problems that require finding solutions over a complex, high-dimensional, and multimodal search space (Eiben & Smith, 2015). Although this does not ensure the global optimum is found, high-quality solutions are frequently obtained in an order of magnitude higher than with standard methods of optimization, particularly when there is little information about the problem (Goldberg, 1989; Michalewicz, 1996).

The next sections give an overview of several application areas in which EAs have been applied, thus showing their adaptability and impact.

## 6.1 Engineering optimization (structural design, control)

Evolutionary algorithms (EAs) comprise a proven and common family of methods used for optimization problems in engineering design, especially in multicriteria structural design (Hajela & Lin, 1992). Many engineering design problems are multi-objective and requires trade-offs to be made among different stresses, weight, aerodynamic drag, costs, and other constraints, making them ripe for population-based search (Deb, 2001; Deb et al., 2002; Coello, Lamont & Van Veldhuizen, 2007). In studies based on the Navier-Stokes equations, Genetic Algorithms have been employed to optimize airfoil and wing geometry, achieving drag reductions while meeting aerodynamic constraints (Obayashi & Takanashi, 1996; Masters & DeLaurentis, 2001). GAs and related evolutionary approaches have been notable successes at truss design under stress and displacement constraints that minimize weight (Rajeev & Krishnamoorthy, 1992; Camp, Pezeshk & Cao, 1998); a comparison of methods for structural optimization also acknowledges simulations based on evolutionary strategies (Baumann & Kost, 1999).

In control engineering, EAs can adjust controller parameters (for example, proportional-integral-derivative regulators) to adapt them for dynamic processes, obviating the need for accurate mathematical models (Michalewicz, 1996). These are conflicting goals, and explorations across a broad range of potential configurations by EAs help stabilize the balancing act so that controllers become more robust and reliable (Eiben & Smith, 2015).

## 6.2 Machine Learning and feature selection

Evolutionary algorithms are often used in machine learning to optimize either feature selection or model hyperparameters (Chandrashekar & Sahin, 2014). In contrast, Genetic Algorithms (GAs) can yield an efficient solution with maximum predictive power without any computational cost or risk of overfitting (Eiben & Smith, 2015). The search space in finding relevant features is typically too ample to try every combination of input features (e.g., exponentially prominent in many cases), and so an exhaustive approach is impossible. GAs can be successfully applied to a variety of classification problems, including medical diagnosis and text categorization, as they iteratively improve a population of subsets of candidate features (Goldberg, 1989; Michalewicz, 1996; Whitley, 1994).

Apart from feature selection, evolutionary algorithms have also been used for hyperparameter tuning on a broad range of techniques (Bäck, Fogel & Michalewicz, 1997; Eiben & Smith, 2015). Practically speaking, the evolutionary hyperparameter tuning helps to bypass various trial-and-error steps that can lead to more accurate models with respect to accuracy under existing resource constraints (Chandrashekar & Sahin, 2014).

## 6.3 Operations Research (scheduling and routing)

Operations Research (OR) brings up many challenging problems, lots of which can be modeled as discrete optimization problems (Gen & Cheng, 1997). Two particularly well-studied examples are:

- *scheduling, manufacturing tasks, and resource allocation*: EAs are widely used for scheduling tasks and resources in manufacturing, computing clusters, or transportation systems (Michalewicz, 1996). Genetic Algorithms are able to fully explore feasible schedules that minimize makespan or maximize resource utilization by representing schedules as permutations or using special encodings (Eiben & Smith, 2015). A comprehensive survey for examination timetabling – covering heuristics, metaheuristics (including GAs), and full system developments – is provided by Qu et al. (2009).
- *routing tasks*: the traveling salesman problem and vehicle routing problem are well-known examples of NP-hard problems (Goldberg, 1989). Known as memetic

algorithms that combine GA with local search heuristics, they often yield high-quality solutions on large-scale instances (Merz & Freisleben, 1999).

Given that EAs are inherently parallel and able to hybridize, they are a natural candidate for use in OR problems of unprecedented scales and complex constraints.

## 6.4 Robotics and autonomous systems

Many researchers in the field of robotics have also applied evolutionary algorithms to evolutionary morphology and controllers. One of the earliest works in evolutionary robotics was the use of Genetic Algorithms (GAs) to evolve finite state controllers and neural networks for agents capable of guiding themselves and dealing with unknown environments (Nolfi & Floreano, 2000). Evolutionary algorithms simulate robotic behaviors over thousands of generations to discover control strategies robust to noise and hardware variations (Bongard, Zykov & Lipson, 2006).

Proponents have thus begun to stress the idea of co-evolving robot morphologies and control systems simultaneously, as new forms of fabrication technology (e.g., flexible manufacturing techniques and 3D printing) allow us to manufacture such creations (Eiben & Smith, 2015). This is a primitive example, but it proves the concept of "bricolage" for designing dedicated robots suitable for some applications like planetary exploration, search and rescue operations, or precision agriculture (Nolfi & Floreano, 2000).

## 6.5 Design of Neural Network architectures

Neural networks, and particularly deep architectures, are highly complex systems to design and tune, since the group of possible designs is astronomically huge (Bäck, Fogel & Michalewicz, 1997). While gradient-based methods (e.g., backpropagation) have been successful in learning the depths of networks and weights between neurons, they are less effective in optimizing network structures (e.g., number of layers, connectivity patterns, types of layers).

Evolutionary approaches like genetic programming (GP) and neuroevolution methods address this problem by searching over the network architecture space directly (Minerva, Paterlini & Poli, 2000; Floreano, Dürr & Mattiussi, 2008). For instance, methods such as NeuroEvolution of Augmenting Topologies (NEAT) enable the evolution of larger and larger neural structures while keeping proven traits intact into new generations (Deb, 2001; Stanley & Miikkulainen, 2002). One of the potential benefits of this is that these methods could reveal new and non-standard architectures that perform better in, e.g., reinforcement learning (Miikkulainen et al., 2017), or system control (Eiben & Smith, 2015).

## 6.6 Application in education and Learning Analytics

Evolutionary algorithms have been increasingly applied in education and learning analytics over the past years (see Figure 6.1 for a schematic view). Learning paths, resource allocation, and curriculum design are central challenges in educational contexts that need to be tailored individually by all within constraints such as limited duration, individualized student requirements, and scarcity of instructors (De Santis & Minerva, 2026; Boyer & Veeramachaneni, 2015; Ma et al., 2014). In the e-Learning domain, evolutionary algorithms (EAs) could be used to personalize how courses are accessed, either by dynamically changing course content or activity sequence based on user interactions and performance (Huang, Huang & Chen, 2007; Chang & Ke, 2013; Tan et al., 2012).

According to one definition given at the 1st International Conference on Learning Analytics & Knowledge (LAK 2011), learning analytics refers to systematic actions of measuring, collecting, analyzing, and reporting data about learners and education contexts with the aim of understanding and enhancing learning processes and environments (Ferguson, 2012). EAs have been used in this area to improve classifiers and clustering models for identifying at-risk students, discovering underlying patterns of student interaction, and enhancing learning material recommendation systems (Huang, Huang & Chen, 2007; Chang & Ke, 2013; Tan et al., 2012; Baylari & Montazer, 2009; Michalewicz, 1996; Khare, 2015). Through the feature sets or model parameters, EAs can evolve iteratively, and by performing such operations, these systems can navigate vast spaces of search and also give insights for educators and administration. Applying intelligent tutoring systems (ITS), Genetic Algorithms can also be used to personalize the level of difficulty in problem-solving tasks or implement adap-

| | Encoding | Objectives | Constraints |
|---|---|---|---|
| Timetabling | Permutation | Makespan / soft penalities | Rooms, teachers |
| Curriculum sequencing | Binary / mixed | Learning gain / workload | Prereqs, max load |
| Item selection | Binary | Reliability vs #items | Max length, coverage |
| Early-warning policy | Binary / mixed | TPR / FPR, cost | Budget, fairness |
| Cohorting | Binary | Silhouette / balance | Size, profiles |

**Figure 6.1** - Optimization targets, objectives and constraints

Network/cluster diagram linking optimization targets in education timetabling, curriculum, predictive models, policies with their objectives and constraints.

tive test questions, showing further support for effectiveness in computer-based learning techniques (Ma et al., 2014; Huang, Huang & Chen, 2007; Chang & Ke, 2013).

## 6.7 Statistical modelling

Genetic Algorithms started as simple generic metaheuristics, but have matured into specialized engines for model search within the domain of statistical modeling: here, individuals are competing model specifications (e.g. subsets of predictors, lag orders or link functions), and "fitness" (some of them are discussed in Table 6.1) is adjudged through an information criterion, likelihood or out-of-sample risk function (Paterlini & Minerva, 2010a, 2010b; Minerva & Poli, 2001a, 2001b; Minerva, Paterlini & Poli, 2000; Pattarin, Paterlini & Minerva, 2004; Whitley, 1994).

With the emergence of new research conducted in recent years using parallel/bagged Genetic Algorithms that enabled giving objective thresholds on GA-based feature "importance" score, more principled selection rules are now possible instead of just arbitrary cutoffs for enhancing reproducibility in model selection workflows (Plante, Larocque & Adès, 2024). In applied mixed-effects modeling, GA-based

**Table 6.1** - Objective functions for GA-based model search and structural selection

Comparison of AIC, AICc, BIC, cross-validation risk, and Pareto fit-size objectives with plain interpretation, complexity penalization, and small-sample suitability. The table clarifies when to favor parsimony-oriented criteria versus direct out-of-sample risk, and recommends reporting of knee points when using multi-objective search. Abbreviations: CV = cross-validation.

| Objective function | Short description | Penalizes complexity | Notes |
|---|---|---|---|
| AIC | Trades fit vs. parameters (2p penalty) | Mildly | Can overfit in small samples; compare models on same data split |
| AICc | AIC with finite-sample correction | Yes (stronger) | Prefer to AIC when n/(p+1) is small |
| BIC | Approx. Bayes criterion with log(n)·p penalty | Strongly | Favors parsimony; consistent under regularity |
| Cross-validation risk | Direct estimate of out-of-sample error | Indirect (via CV) | Use k-fold or repeated CV; report variance across folds |
| Pareto fit–size | Joint optimization of predictive fit and sparsity | Explicit | Use NSGA-II; report knee points and selected compromise model |

approaches that target problems specific to a covariate model for the management of collinearity and introduction of diverse initial populations generally reduce the convergence time frame, leading to fewer redundant terms, which – in the reported case studies – give better AIC values than stepwise SCM (Ronchi et al., 2024).

To address the time series identification problem, a hybrid GA-SARIMA approach was proposed, which searches for SARIMA (Seasonal ARIMA = Autoregressive Integrated Moving Average) orders in parallel using a single-objective genetic algorithm to avoid local minima in Box-Jenkins framework and thus increase forecasting accuracy for large, noisy datasets of time series (Farsi et al., 2021). In addition to using GAs for structural search, Genetic Algorithms have also been employed as optimizers when error distributions are heavy-tailed, or the likelihood function is multimodal, and when GAs can outperform traditional iterative methods for multiple-regression maximum-likelihood estimation (Yalçınkaya et al., 2021). Together, these results confirm that Genetic Algorithms represent a complementary methodology to penalization and exact mixed-integer optimization; they provide an efficient mechanism for exploring combinatorial model space, identifying Pareto (efficient) compromises between fit and parsimony that serve as a stepping stone to conventional estimation and post-selection inference.

# Chapter 7

# Advanced topics

New strategies and extensions have been developed as evolutionary algorithms (EAs) evolved over the years to cope with more complex and larger-scale problems. This section presents four sides of the advanced edge: multi-objective optimization (section 7.1), constraint handling (section 7.2), coevolution (section 7.3), and parallel/distributed evolutionary algorithms (section 7.4).

## 7.1 Multi-objective Evolutionary Algorithms (MOEAs)

Many real-world problems concern optimizing multiple objectives (also termed criteria) related to the functions of a solution, which, however, often conflict in the sense that there is no single solution that is optimal across all these objectives (Coello, Lamont & Van Veldhuizen, 2007). Multi-objective evolutionary algorithms (MOEAs) approximate the Pareto set, typically maintaining an elite archive of non-dominated solutions (Deb, 2001, 2011).

### Pareto Dominance

Solution A dominates solution B if A is at least as good as B in all objectives and strictly better in at least one. The main goal of MOEAs is to approximate a well-distributed set of non-dominated solutions (i.e., the Pareto-optimal front), since this represents the balanced trade-offs between two or more conflicting objectives (Deb, 2001, 2011).

In Figure 7.1, we show an example of a two objective plot with the corresponding Pareto front.

### Representative Algorithms

Classic MOEAs include NSGA-II (Deb et al., 2002) and SPEA2 (Zitzler, Laumanns & Thiele, 2001). These algorithms use specialized selection strategies, diversity necessitation, and elitism strategy to efficiently approximate the Pareto front.


**Figure 7.1** - Pareto Front in Multi-objective Optimization
Two-objective scatter plot showing non-dominated solutions, knee point, and crowding distance. It visualizes the concept of Pareto efficiency used in NSGA-II and related MOEAs.

### Applications

For their general effectiveness on problems involving multiple objectives, MOEAs have been applied to classical tasks in many fields (engineering design, portfolio optimization, and scheduling) where decision-makers must juggle trade-offs among a number of conflicting objectives (Coello, Lamont & Van Veldhuizen, 2007; Deb, 2001, 2011).

## 7.2 Constraint handling in Evolutionary Algorithms

Most real-world optimization tasks include complex constraints (see Figure 7.2) like capacity, security, and resource limitations that can make naïve solutions infeasible

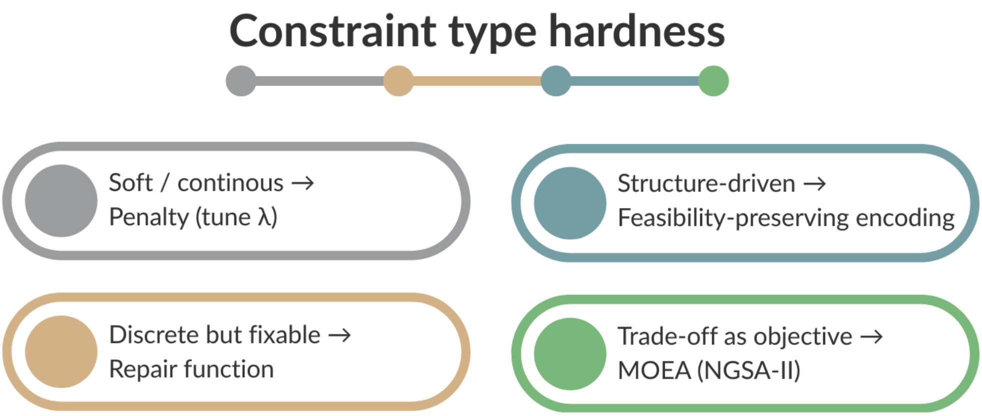


**Figure 7.2** - Constraint-handling strategies in EAs
Diagram comparing penalty, repair, feasibility-preserving, and multi-objective strategies for handling constraints, arranged along a continuum of constraint hardness.

(Michalewicz & Fogel, 2004). However, designing EAs to handle constraints often relies on one of the following strategies:

- *penalty methods*: when infeasible solutions are generated or present in the entire population, their fitness evaluation adds a penalty to the objective proportional to the degree of violation (Michalewicz & Fogel, 2004).
- *feasibility-first*: the algorithm either destroys or fixes the infeasible solutions, keeping that entire population always within normal bounds. This method is the most obvious and straightforward, but it could lessen the diversity (Deb, 2001, 2011).

### Multi-objective reformulation

Constraints are treated as additional objectives to minimize or satisfy in a multi-objective evolutionary algorithm (MOEA) framework, maintaining the balance between feasibility and other objective functions (Coello, Lamont & Van Veldhuizen, 2007).

### Repair operators

Whenever the operator gives an infeasible solution, a repair function is conducted to fix the violations based on the nature of the problem. Genetic diversity can be maintained when freezing a diverse subset (Michalewicz & Fogel, 2004).

Choice of an adequate constraint handling strategy depends on the problem complexity and also the nature of constraints (hard or soft) (Coello, Lamont & Van Veldhuizen, 2007).

## 7.3 Coevolution and competitive environments

In some domains, solutions interact with one another – either cooperatively or competitively – and require special adaptations, called coevolutionary algorithms.

Two forms of coevolution have been proposed: in a *cooperative* type, different subpopulations evolve together towards composing a complete solution (Potter & De Jong, 2000), while in a *competitive* form, the individuals from different subpopulations play against each other also using game theoretical perspective as in Rosin & Belew (1997).

In detail:

- in *cooperative coevolution*, subpopulations encode subcomponents; individuals are jointly evaluated via team formation, promoting specialization and robustness (Potter & De Jong, 2000; Eiben & Smith, 2015);
- in *competitive coevolution,* solutions evolve against each other, e.g., in predator-prey scenarios or games (e.g., chess, Nim, Tic-Tac-Toe) (Rosin & Belew, 1997). This results in increasingly advanced tactics to emerge because each individual is always trying new strategies and solutions against a moving target defined by other evolving agents.

Cooperative coevolution is advantageous when a natural decomposition exists; competitive coevolution is appropriate for adversarial/dynamic tasks (Eiben & Smith, 2015).

## 7.4 Parallel and Distributed Evolutionary Algorithms

The iterative and population-based structure of evolutionary algorithms (EAs) can be exploited with parallel and distributed implementations to provide a substantial computational advantage, especially in the case of high-dimensional problems (Alba & Tomassini, 2002; Cantú-Paz, 2001; Biscani & Izzo, 2020; Izzo, 2012).

Common approaches include (a schematic plot is shown in Figure 7.3):

- *Master-Slave Model*: in this model, there is one controller node that controls the population and dispatches fitness evaluations to several agent nodes to use parallel processing (Cantú-Paz, 2001).
- *Island (Distributed) Model*: it partitions the population into semi-isolated subpopulations, that is, "islands" that evolve mostly independently; occasional migration exchanges individuals, improving scalability and reducing premature convergence (Alba & Tomassini, 2002; Cantú-Paz, 2001; Biscani & Izzo, 2020).
- *Cellular (Diffusion) Model*: in this context, individuals are located in a 2D (and even ND-) grid and interact only with their neighbors. So, these localized interactions slow down the genetic drift and maintain a diversity level (Alba & Tomassini, 2002).

Contemporary computing infrastructures, such as multicore processing units (CPUs), graphics processing units (GPUs), cluster and cloud environments, have been considerably adopted in parallel EAs, making these types of EA particularly well-suited to computationally intensive tasks (Eiben & Smith, 2015; Michalewicz, 1996).

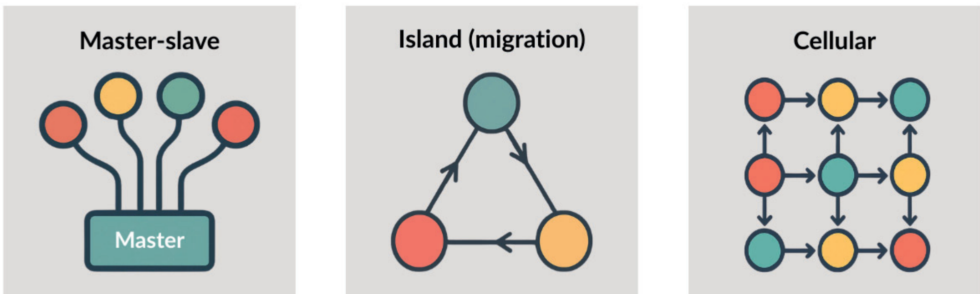


**Figure 7.3** - Parallel EA architectures

Three small schematics: star (master-worker), clustered nodes with migration arrows (islands), and a lattice (cellular EA).

# Chapter 8

## Current trends and research directions

Over the years, thanks to research into the limits of population-based approaches, evolutionary algorithms (EAs) have undergone rapid changes. We review below four key research areas in modern evolutionary computation (EC). In detail: theoretical foundations, hybrid metaheuristics, real-world optimization at large scale, and interpretability (Eiben & Smith, 2015).

### 8.1 Theory of Evolutionary Computation (time complexity, convergence)

Initial works on EAs were motivated by practical needs. This interest, however, was soon shifted to the study of their performance from a theoretical perspective or model-based analysis (see Auger and Doerr, 2011). Key theoretical topics include:

- *time complexity*: studies report the time required for EA to reach near-optimal solutions (Neumann & Witt, 2010). This kind of analysis is usually performed through means like drift analysis and Markov-chain arguments to measure contractors' algorithmic performance across generations (Droste, Jansen & Wegener, 2002).
- *convergence properties*: theoretically, convergence proofs (e.g., Rudolph, 1997) can specify conditions that guarantee global convergence in the long term. At the same time, preliminary convergence is a practical problem that can be addressed separately by selection pressure, mutation rate, or population size.
- *landscape analysis*: scaling fitness landscape analysis allows to systematically quantify problem difficulty and guide the design of an optimization algorithm (Pitzer & Affenzeller, 2012). Landscape-based methods are able to capture the structural properties of the search space, which guides evolution towards better solutions and helps design appropriate parameter tuning and operator selection.

It has laid the groundwork for a more robust theory of evolution strategies, which helps ensure that the proposed algorithms adhere to best principles and can be meaningfully compared (Auger & Doerr, 2011). See also Beyer & Schwefel (2002) for Evolutionary Strategies theory.

## 8.2 Hybrid Metaheuristics

Hybrid metaheuristics combine evolutionary algorithms with many other techniques, such as local search heuristics and/or sophisticated machine learning models (Glover & Kochenberger, 2003). Common motivations for hybridization include:

- *local-search intensification (memetic hybrids):* utilizing local implementations, such as gradient-based steps or specific heuristics that are either internal to the search (local genetic operators) or in the outer scope of the optimization problem, can improve solution quality and reduce iterations (Hart, Smith & Krasnogor, 2005).
- *complementary strengths*: the combination of EAs with deep learning (neuroevolution) or reinforcement learning brings together the global search capabilities of EAs and the representational power of neural networks (Floreano, Dürr & Mattiussi, 2008; Moscato, 1989).
- *multi-stage pipelines*: in large-scale data analytics or knowledge discovery, the EA may be used for selecting a subset of features/hyperparameters, and subsequent machine learning techniques are employed to handle model training/fine-tuning (Chandrashekar & Sahin, 2014).

With computer power increasing and real-world problems becoming more complex, the interest in hybridization of metaheuristics is unlikely to fade away as it remains one of the research spotlight in metaheuristic research (Eiben & Smith, 2015).

## 8.3 Large-scale and real-world optimization problems

Optimization problems in some areas and disciplines, such as bioinformatics, product delivery through the internet, and climate modeling, face ever-increasing challenges with modern optimization sites, including very high-dimensional data and massive instance sizes (Escalante, Montes & Sucar, 2015). The ability to address such large-scale or highly complex problems is beyond the technical sophistication of conventional evolutionary algorithm (EA) designs:

- *scalability*: research into scalable representations and operators, methods of decomposition, or surrogate modeling to allow for efficient traversal of high-dimensional spaces (Alba & Tomassini 2002; Biscani & Izzo, 2020; Jin, 2011).
- *parallel and distributed computing*: parallel EAs that use GPUs, high-performance computing (HPC) clusters, or cloud platforms allow the acceleration for large populations and more frequent evaluations (Alba & Tomassini, 2002; Biscani & Izzo, 2020; Izzo, 2012).

This situation arises because real-world problems tend to be dynamic and often involve uncertain data or changing constraints over time. Real-time Evolutionary Algorithms, which remain an active area of research, address this challenge through continuous schedule updates (Nguyen, Yang & Branke, 2012).

The development of EAs that are tailored to the scale and nature of these challenges is necessary for improving EAs' ability to address industrial and societal problems.

## 8.4 Interpretable Evolutionary Models

Interpretability or explainable AI is even more required in the time of complex, or black-box models such as deep neural networks, including Large Language Models (see, e.g., Molnar, 2019). Many ways are being studied within the realm of evolutionary computation to make ML models interpretable.

- *Genetic Programming (GP):* GP evolves expression trees that are human-readable and whose evaluation provides a direct function from the input to the output (Koza, 1992). More recent work has attempted to manage the size or complexity of trees (Tran, Xue & Zhang, 2019) in an effort to achieve a trade-off between predictive power and interpretability.
- *rule-based methods:* Evolutionary rule induction yields IF-THEN sets that are comprehensible (Freitas, 2014); see also Chakraborty (2014) for evolutionary rough-fuzzy attribute reduction.
- *local explanations for EA-optimized models:* Evolutionary Algorithms (EAs) may produce models that are too complex for direct interpretation, in which LIME-like or SHAP-like methods can provide a local post-hoc explanation (Molnar, 2019). This preserves the performance gain of complex representations while also addressing the need for transparency with stakeholders.

With AI systems proliferating, the requirement that evolutionary solutions must work and be comprehensible to human users and regulators is likely to become a fundamental research question (Eiben & Smith, 2015).

# Chapter 9

## Summary and concluding remarks

This first part has given a general sense of evolutionary algorithms (EAs), starting from inspiration in biology, then moving on to the basic working mechanisms of the Genetic Algorithm (GA) – representation, initialization, fitness assignment, selection, crossover, mutation, and stopping procedures. These are followed up by more generic Evolutionary Computation (EC) paradigms, including Genetic Programming (GP), Evolutionary Strategies (ES), Evolutionary Programming (EP), Differential Evolution (DE), and Memetic Algorithms.

The implementation considerations were discussed, with a focus on parameter tuning, computational complexity, and popular toolkits available in different programming languages. We, then, addressed the same topic with respect to different fields like engineering, machine learning, operations research, robotics, statistical modeling, and neural network architecture design as concrete EAs. Advanced topics like multi-objective optimization, coevolution, and Parallel EAs provided us with a sophisticated idea of the capabilities of EC techniques. Lastly, current developments are instantiated in the changing theoretical basis of EAs, the era of hybrid metaheuristics, big-scale real-world optimization, and the focal point of in-depth comprehension.

As optimization algorithms, evolutionary algorithms have shown remarkable adaptability and efficiency, mimicking the basic principles of natural selection and genetic variation. Their widespread applicability, along with a large, active community of theoretical and applied researchers keeping indoor problems at the state-of-the-art, will keep EAs at the heart of contemporary optimization and AI projects. With computational resources growing and cross-disciplinary applications appearing, the power of evolutionary computation to tackle complex problems has never been stronger.

# PART II

# GENETIC ALGORITHM AND EVOLUTIONARY COMPUTATION IN LEARNING ANALYTICS AND EDUCATION: METHODS, APPLICATIONS, EXAMPLES, AND CASE STUDIES

# Chapter 10

## Introduction

The scene of modern education is transforming with data-driven technologies being applied to address some traditional sticking points around student outcomes like personalized learning, resource allocation, and strategies for accurate early identification of students at risk. In this domain, Genetic Algorithms (GAs) and Evolutionary Computation (EC) have proved to be effective approaches for complex optimization in educational situations, allowing controlled exploration of large constrained spaces that are often non-differentiable. At the same time, Large Language Models (LLMs) have quickly pervaded educational technology as tools for generating feedback, authoring, and analytic workflows.

This section discusses how GAs and EC, as population-based probabilistic search methods, accommodate both traditional statistical learning and LLM-enabled analytics approaches in Learning Analytics (LA) and Educational Data Mining (EDM).

As LA and EDM have matured – powered by the digitization of learning environments and the resulting surge in behavioral, assessment, and interaction data – standard modeling pipelines have expanded from classical statistical methods to a spectrum of machine learning, deep learning, and generative approaches. However, the complex nature of educational systems (De Santis, 2022) typically goes beyond the power of deterministic approaches per se: the problems are high-dimensional and combinatorial (e.g., test assembly, timetabling, pathway sequencing), non-smooth (e.g., fairness constraints, exposure limits, access and cost considerations), multi-objective (e.g., accuracy vs. interpretability; predictive power vs. stability; learning gain vs. equity). GAs and EC are designed for this domain: they preserve populations of candidate solutions and adapt through variation (crossover, mutation) and selection across iterations, naturally exploring Pareto-optimal tradeoffs when objectives conflict.

Together with the applications of algorithmic solutions, higher education institutions are focusing (or should focus) on data analytics as an area for facilitating aligned decision-making across teaching and learning, as well as administration (Nguyen, Gardner & Sheridan, 2020). The field of analytics in education falls under a broader paradigm shift toward smart learning environments, which stresses adaptivity and personalization (Kinshuk et al., 2016).

This view also makes clear why LLMs and EC are complementary technologies rather than competing ones. LLMs are trained with gradient-based optimization over a massive scale, and, at inference time, they can reasonably be considered as conducting probabilistic search over sequences given constraints on such sequences (prompts, decoding strategies, safety rules). In practice, the agentive aspects of LLMs – representation and synthesis, heuristic generation (e.g., evidence summaries, plausible constraint proposals, template rubrics or item stems, scaffolding analytic narratives) – are complemented by EC in a role formalizing search and optimization in the outer loop where objectives are discrete or non-differentiable (e.g., blueprint compliance in test assembly; fairness constraints in early-warning models; schedule-validity measures such as makespan in timetabling). To put it succinctly, LLMs expand what we can express and prototype, while EC formalizes how we search and decide given a set of explicit objectives and constraints.

This Part explores the potential of leveraging GAs and EC (and, occasionally, their integration in LLM-assisted workflows) to improve educational outcomes and operational efficiency across LA and EDM. We focus on the following topics:

1. *personalization:* how can GAs and EC optimize sequencing, prerequisites, and workload under multiple goals (e.g., mastery gain, fatigue limits, equity) to create learning pathways that will fit diverse learners?
2. *design and operations:* in what ways can these algorithms optimize curriculum structure, course and exam timetables, and assessments while taking into account constraints related to content, difficulty, and exposure?
3. *predictive modeling:* how do EC methods support model selection, feature construction, and early identification of at-risk students for promoting engagement in online learning, especially when objectives (accuracy, sparsity, fairness, interpretability) conflict?
4. *LLM-EC synergy:* where do LLMs accelerate problem formulation (e.g., constraint elicitation, policy drafting, feedback rubrics) while EC is performing decision-grade optimization (e.g., selecting a balanced set of items or interventions hitting explicit targets)?

In addressing these questions, we explicitly consider scalability, computational budgets, and ethical obligations, given that educational data can carry latent bias, privacy risks, and access disparities. We therefore discuss safeguards such as defining fairness-aware objectives and constraints, adopting robust cross-validation protocols, and transparently reporting trade-offs along the Pareto front. In addition, when LLMs are enabled in workflows, we prioritize applications that support solid human oversight and task provenance, with high-fidelity specification of decision criteria, ensuring that generative assistance augments formal evaluation and optimization rather than displacing them.

Methodologically, this second part proceeds with a critical review and synthesis of the literature alongside representative case studies and well-defined examples. We examine under what scenarios EC methods show clear advantage over standard heuristics or pure gradient-based solvers; we describe their drawbacks (e.g., fitness evaluation cost, tuning sensitivity to encoding and/or operators, premature convergence issues); we present practical guardrails (constraint handling via penalties/repairs/feasible encodings; multi-objective algorithms; reproducibility and stability checks; post-selection inference when relevant). Ethical challenges – algorithmic bias, differential privacy, accessibility, and explainability – are treated as core design constraints, not afterthoughts.

Several different applications of EC to education have already been documented in the existing literature: adaptive content sequencing; item/test assembly; course timetabling; feature selection for predictive modeling; and behavioral clustering of learner traces. Studies and pilots have shown that EC can produce better solutions when objectives conflict or constraints are tight. The computational requirements and scaling that need to be achieved are still open questions. They will be discussed together with several means of reducing this demand, such as surrogate modeling, caching fitness constituents, and parallel evaluation.

To guide the reader, the remainder of Part II is organized as follows. Chapter 10 reintroduces the principles of GAs and EC, showing their use in educational settings in general terms, and introducing applications in the fields of Learning Analytics and Educational Data Mining. We contrast these methods with deterministic optimization and propose connections to LLMs. Chapter 11 examines applications in adaptive learning systems, curriculum design, resource allocation, early-warning modeling, and intelligent tutoring systems, grounded in research and realistic scenarios. Chapter 12 proposes case studies and implementations of GAs in education landscape retrieved from scientific literature focusing on curriculum personalization and optimization, group formation, resource recommendation and adaptive paths, timetabling and resource allocation. Chapter 13 covers EC in educational data mining, including the use of GAs for feature selection, classification, clustering and rule generation. Chapter 14 presents hybrid models, in which EC is combined with machine learning and deep learning (and, where appropriate, LLM-assisted pipelines) to balance predictive performance with interpretability and governance. Chapter 15 deals with ethical and practical challenges – scalability, reproducibility, fairness, privacy, accessibility – along with reporting standards for Pareto trade-offs and constraint satisfaction. Chapter 16 explores new opportunities, suggesting applications of those techniques in lifelong learning and workforce development. It discusses the potentials of LLM-assisted design and EC-driven optimization to promote equitable access, transparent decision processes, and accountable innovation in education.

In summary, this part argues that optimization is a first-class concern in education, that GAs and EC give the right abstractions for exploring rich decision spaces, and that LLMs, properly governed, can accelerate the formulation and communication of those problems while leaving the choices to be settled for transparent objective-directed optimizations.

## 10.1 Overview of Genetic Algorithms (GAs) and Evolutionary Computation (EC) in education and combination with LLMs

In the last decades, Genetic Algorithms (GAs) and the broad class of Evolutionary Computation (EC) approaches have demonstrated the ability to tackle a wide range of complex optimization problems inherent in educational systems.

Such an evolutionary approach to search is well-suited to the nature of most learning environments, where objectives may be in tension (e.g., quality of personalization of students' path versus effort for staff), constraints are many and varied (planes available, instructors on hand, students who show up), and solution spaces are combinatorial. When this last assumption does not hold, the fact that EC can search across the entire optimization surface while honing in on promising areas might offer a much more appealing option or supplement to deterministic optimization. This is manifest in the ability of GAs to automatically organize syllabi into coherent sequences respecting prerequisites, educational goals and pacing, tiptoe around scheduling conflicts in timetables, and design interventions that maximize a tradeoff between accuracy, cost, and fairness – tasks for which linear programming (or greedy heuristics) struggle when constraints or objectives arise as heterogeneous or dynamic.

Classical probabilistic machine learning frameworks (Bishop, 2006; Murphy, 2012) focused almost exclusively on gradient-based optimization under well-specified likelihood models, whereas evolutionary computation provides a derivative-free, population-based alternative suited to discrete and multi-objective educational problems (see Figure 10.1 for a comparison).

In the next pages, we describe areas of application of GAs that will be presented in more detail through case studies in the next chapters, and explain the combination and differences between this class of algorithms and LLMs.

### Curriculum design and sequencing

Curriculum design problems – deciding what to learn when, given prerequisites, time budgets, and individual learner goals – are nonlinear, combinatorial, and frequently multi-objective (e.g., maximize mastery and motivation while minimizing time-to-competency). In cases such as curriculum sequencing, a GA can encode the candidate solution in terms of a list of modules (micro-units) and evaluate fitness combining prerequisites satisfaction, potential learning gain estimation, and practical constraints, e.g., lab availability. Mutation and crossover allow exploring alternative orderings while preserving functional "building blocks" (e.g., tightly coupled module clusters). The same procedure extends naturally to adaptive sequencing. As new information about learner progress arrives (e.g., questionnaire scores), the fitness function can weight mastery and engagement signals containing concept drift, and, as a consequence, the search can favor sequences that maximize mastery per unit time. Because

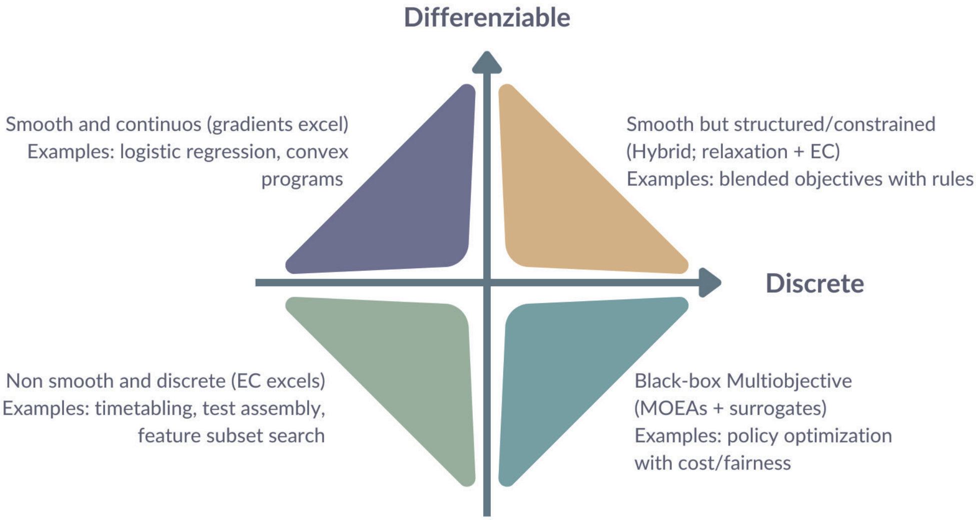


**Figure 10.1** - Method fit across problem structure

Typology matrix mapping problems to methods by smoothness/structure: where gradients shine, where EC dominates, and where hybrids are preferred.

this is a combinatorial space, population-based evolutionary algorithms are useful because they produce multiple valid and different solutions in a single run, avoiding forcing teachers and educators to choose a single "optimal" but fragile solution. Multi-objective evolutionary algorithms (MOEAs), particularly NSGA-II (Deb et al., 2002), explicitly preserve a Pareto front, balancing conflicts (for example: maximize predicted learning gain, minimize time-on-task, and limit instructional cost), requiring a reflection of more realistic relations involved in academic decision-making.

### Resource allocation and timetabling

Classes, exams, and student timetabling problems are classical NP-hard problems where GAs excel by encoding schedules and applying repair operators. They enforce hard constraints and use soft-constraint penalty reduction for compactness, fairness, and preference satisfaction. Preference-aware genetic timetablers can scale to tackle the real-world volume of enrollments while reducing a significant number of conflicts through a multi-objective approach. MOEAs, in fact, can balance equity and utilization (e.g., similar schedule quality across cohorts) and provide leaders with clear tradeoff surfaces rather than a single number. Surveys on educational timetabling (Ceschia, Di Gaspero & Schaerf, 2023; Babaei, Karimpour & Hadidi, 2015) also document standard benchmarks in the literature, revealing where evolutionary search offers practical benefits.

### At-risk identification and engagement modeling

Multimodal traces from the virtual domain, like clickstreams of learner interactions with VLEs, log files of scores on formative assessments, and demographic data, have informed early-warning models. Deep models trained on VLE big data show strong baseline accuracy; EC complements these by choosing compact subsets of high signal features and tuning the operating threshold to optimize local goals (e.g., earlier alerts but acceptable trade-off on false positives). These pipelines treat the alerting problem as a multi-objective optimization task to improve recall for at-risk learners without burdening advisors with false leads (as is common for simple thresholding functions) in applied studies.

### Intelligent tutoring and adaptive platforms

In Intelligent Tutoring Systems and at-scale platforms, EC can optimize content selection, pacing, and hinting strategies against competing goals, just like learning gain, engagement, and cost. In these applications, multi-objective optimization is again important for tailoring teaching approaches and methods to different kinds of students: one cohort may benefit from slightly longer time spent on the task if it leads to greater mastery gains; another may require stringent time limits to maintain motivation. By revealing a Pareto set, EC allows educators to select policies compatible with their context and values, as well as modify them when evidence arises. NSGA-II appears to be a faster practical solution than other algorithms, which works on elitism and well-understood behavior.

### Where LLMs fit and why EC still matters

LLMs are typically trained or fine-tuned via gradient methods. However, at deployment time in education, they create new discrete design spaces – prompts, tool-use programs, rubric templates, feedback styles – that resist straightforward differentiation. Here, EC and LLMs may be seen as complementary (see Figure 10.2):

1. *prompt and policy optimization:* recent work demonstrates that LLMs can be utilized as optimizers through "Optimization by Prompting" (OPRO), and that specific evolutionary algorithms (EvoPrompt, Promptbreeder) effectively mutate and recombine prompts or tool-use programs to maximize performance on downstream tasks. In LLM-based tutors, that manifests as prompts for explanation, Socratic questioning, or misconception repair; in support of grading assistants, evolving up rubric-style text and exemplars to increase agreement and fairness. EC's population maintains diversity – important when the performance landscape has many local optima – while assessment continues to be linked to educational metrics (rubric agreement, learning gain, student satisfaction) (Guo et al., 2024; Yang et al., 2024; Fernando et al., 2023).
2. *co-optimization around LLMs*: LLMs can generate candidate interventions (e.g., activity variants, formative questions) and EC can select and evolve the best ac-

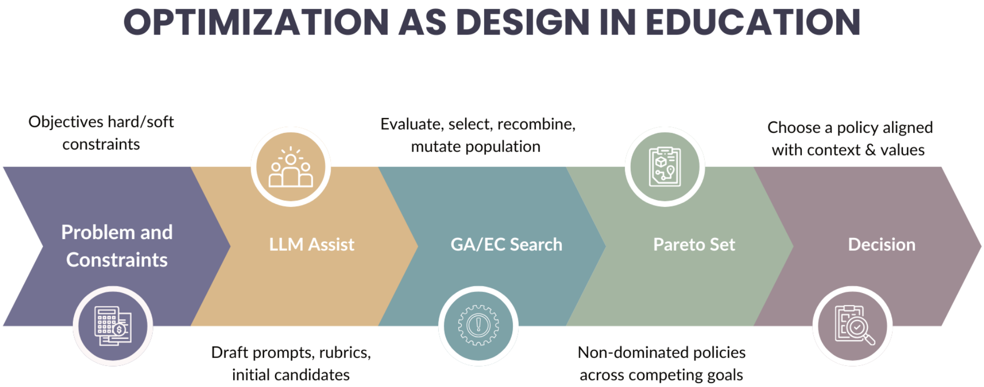


**Figure 10.2** - Optimization-as-Design in education
Complementarity between LLM-assisted proposal and EC-driven optimization. The pipeline highlights constraints/objectives, GA/MOEA search, Pareto-set exposition, and decision selection.

cording to real student responses, closing the loop between generative creativity and measured impact. Conversely, EC can search around LLMs choosing tools, few-shot exemplars, or chain-of-thought templates without retraining the model.

## Why EC over purely deterministic methods?

Educational problems typically involve a combination of discrete and continuous choice, hard and soft constraints, and multiple objectives that cannot be credibly collapsed into a single scalar, as there are too many dimensions in which they differ. Gradient-based methods excel when objectives are differentiable and convex. However, many educational design choices (course orderings, class assignments, prompt templates) are discrete, non-convex, or non-differentiable. EC handles these natively. Moreover, when objectives conflict – accuracy vs. effort, equity vs. utilization – MOEAs provide a set of reasonable solutions for deliberation rather than a single answer, supporting transparent, human-in-the-loop governance. Finally, with LLMs now ubiquitous, institutions need outer-loop optimizers to tune prompts, policies, and pipelines: EC offers that capability, with a track record in combinatorial problems such as timetabling and scheduling.

*Takeaway.* Based on a multi-objective approach, EC provides search and optimization solutions that modern educational systems need: it allows customizing learning paths, tuning predictive pipelines, and addressing major operational challenges, such as resource allocation. These algorithms can work alongside LLM-powered systems with gradient-trained LLMs at the core, evolutionary search at the edges. Educational platforms must iteratively build up from scratch while respecting constraints and exposing tradeoffs to create a round of personalization, fairness, and efficiency benefits.

## 10.2 Relevance to Learning Analytics (LA) and Educational Data Mining (EDM)

The intersection of Learning Analytics (LA) and Educational Data Mining (EDM) with Genetic Algorithms (GAs) and, more broadly, Evolutionary Computation (EC) – or even more statistical and computational techniques – arises from the use of educational online environments and the resulting production of vast, heterogeneous, rapidly changing data that needs to be analyzed and interpreted to obtain advances in teaching and learning processes (De Santis, 2022; De Santis et al., 2024). On the education side, LA and EDM look to use digital traces (clickstreams, assignments, forum interactions, logs) to actionable insights and timely interventions. On the computational side, GAs and EC provide population-based search, feature subset discovery, and multi-objective optimization, which are useful for contrasting educational objectives (e.g., personalization vs. resource limits; early detection of risk vs. minimization of false alarms). In a nutshell, LA and EDM define what to optimize and why (Ferguson, 2012; Long & Siemens, 2011; Papamitsiou & Economides, 2014; Lang et al., 2022); GA and EC, among other techniques, supply ways to do so at scale.

### Predictive modeling for early warning

Central to LA/EDM are tasks for predicting course outcomes, disengagement, or dropout well enough in advance to enable actions. Here, GA/EC are valuable on two fronts. First, they improve feature selection, automatically discovering compact, high-signal indicator sets from a vast space of behavioral and contextual variables (e.g., attendance patterns, LMS logins, quiz latency, forum centrality, prior grades), which often yields better generalization than hand-picked features. Second, evolutionary search tunes model hyper-parameters and directly optimizes decision policies (e.g., timing of alert triggers, prioritization of outreach) under multi-objective criteria such as sensitivity, specificity, and staff workload without assuming smooth objective surfaces. Data-driven pipelines for early risk detection are highlighted in review works in LA/EDM, with some studies demonstrating that feature-selection-driven models can target interventions at scale (Albreiki, Zaki & Alashwal, 2021; Márquez-Vera et al., 2016; Rastrollo-Guerrero, Gómez-Pulido & Durán-Domínguez, 2020; Xue et al., 2016; Xing et al., 2015).

For example, Márquez-Vera et al. (2016) report that evolutionary pipelines can yield better results than conventional baselines for predicting early high-school dropouts by evolving explainable IF–THEN rules that achieve a good compromise on sensitivity/specificity in highly unbalanced data. Their approach proved possible for valid predictions over the first 4-6 weeks of the semester, relying on demographic, behavioral, and early performance indicators.

### From static pictures to adaptive personalization

Beyond prediction, EC supports adaptive learning by continually re-optimizing content, pacing, and assessment based on observed learner trajectories. Unlike static recommendation rules, evolutionary controllers re-balance exploration/exploitation over time, trying new learning objects when confidence is low, consolidating when mastery indicators are strong. In large-scale environments (e.g., MOOC platforms and online universities), this adaptive layer must remain effective under volume (many learners), variety (heterogeneous learners and resources), and velocity (fast streaming data).

### Institutional logistics, timetabling, and resource allocation

LA is not only about students; it also informs institutional operations. As already said in the previous paragraph, timetabling and resource allocation are classic NP-hard combinatorial tasks (rooms, sections, instructors, student groups, constraints). GA-based and hybrid evolutionary timetablers consistently produce feasible, high-quality schedules, reducing conflicts and better respecting preferences, e.g., instructor availability, room capacities, program-specific constraints (Abdelhalim & El Khayat, 2016; Bashab et al., 2023). Recent surveys (Ceschia, Di Gaspero & Schaerf, 2023) on educational timetabling report consistent performance gains from hybrid approaches combining metaheuristics, mathematical programming, and decomposition strategies, especially for large and realistic benchmark instances.

### Hybrid pipelines and multi-objective tradeoffs

LA/EDM deployments today are hybrid, combining various techniques. Speaking of evolutionary strategies as in our case, we can observe that GAs can be used to select features and hyperparameters for downstream learners (e.g., tree ensembles, neural nets), or to allocate weights for stacked models that integrate behavioral, demographic, and contextual predictors. The hybrid approach aligns closely with multi-objective optimization; for instance, jointly maximizing early-warning recall while minimizing false-positive burden on advising staff or learning gains versus content production costs in an adaptive platform.

### Connections to Large Language Models (LLMs)

Large Language Models (LLMs) and Evolutionary Computation (EC) meet in educational applications: LLMs generate discrete, black-box design spaces (e.g., prompts, tool-use programs, rubric templates, feedback styles) whose performance is observable but not differentiable, and EC supplies derivative-free, population-based search that can explore those spaces, conserve diversity, and reveal tradeoffs. In tutoring, grading, advising, and LA/EDM workflows, this pairing enables institutions to optimize LLM behavior against pedagogical metrics (learning gains, rubric agreement, fairness, satisfaction) under real constraints (staff capacity, latency, privacy).

Three scenarios can be described.

1. *EA-driven prompt search*. Evolutionary prompt optimization treats instructions as genomes and uses mutation/recombination to generate candidate prompts, selecting those that maximize a task-specific score (e.g., accuracy, rubric agreement, engagement). Across diverse benchmarks, evolutionary prompt search has outperformed hand-crafted prompts, a result directly applicable to evolving Socratic questioning styles, misconception-repair explanations, and grading-rubric phrasing in LLM-based tutors and assessors (Guo et al., 2024; Fernando et al., 2023; Zhou et al., 2023).
2. *LLMs as EA components or as optimizers.* Some works display LLMs as evolutionary optimizers for combinatorial problems without retraining or domain-specific operators (Guo et al., 2024; Fernando et al., 2023). In education, this allows decision-makers to state constraints in plain language (e.g., avoid overlapping Year-2 labs; respect instructor preferences) while retaining EC's search power for timetabling, adaptive sequencing, or intervention-policy design (Yang et al., 2024; Guo et al., 2024). Equally, adopting evolutionary logic, "Optimization by PROmpting" (OPRO) shows that an LLM can iteratively generate improved candidates by conditioning on prior solutions and their scores, effectively emulating aspects of evolutionary search through natural language prompting (Yang et al., 2024).
3. *Upstream/downstream synergy in LA/EDM pipelines.* Upstream, evolutionary feature selection yields compact, high-signal indicators (attendance volatility, quiz-latency dispersion, forum centrality) that feed LLM-based advising assistants, improving interpretability and reducing data costs. Downstream, free-text rationales generated by LLMs for risk flags or feedback can be scored and refined to maximize consistency, actionability, and fairness over time. This evolutionary "outer loop" complements non-evolutionary prompt methods (e.g., Automatic Prompt Engineer - Zhou et al., 2023; RLPrompt - Deng et al., 2022) when gradients are unavailable or unreliable.

## Ethics, privacy, and fairness

As LA/EDM systems scale, ethical governance becomes a first-order concern involving privacy, consent, purpose limitation, bias mitigation, and the obligation to act on risk predictions. EC adds both opportunities, like explicitly optimizing fairness constraints alongside accuracy, and responsibilities in accountability and transparency in the procedure and choices. Good practice involves documenting the optimization objectives, constraints, and tradeoffs (e.g., Pareto fronts) in language accessible to learners and staff. It is good practice to record the objectives, constraints, and potential tradeoffs (e.g., Pareto fronts) in a human-readable format for learners and staff. Ethically, analytics need to be supportive, not surveillant, and shapes of intervention triggered by predictive models must be fair and transparent (Slade & Prinsloo, 2013).

### Scalability and operations

Finally, GA/EC are well-suited to the operational realities of LA/EDM-distributed computation, streaming data, and periodic re-optimization. Because evolutionary methods evaluate many candidates in parallel, they map naturally onto modern compute clusters and can be scheduled in as-you-go cycles aligned with academic calendars (e.g., re-fit early-warning models after add/drop; re-opt timetables when sections overflow). Moving from descriptive dashboards to optimization-aware services is a necessary step for analytics to have a material impact on student success and institutional efficiency.

*Takeaway*. LA and EDM describe the *what* (better predictions, earlier interventions, more personalized pathways, and most efficient operations. GA and EC produce the *how* under real-world complexity: noisy high-dimensional data, conflicting goals, dynamic constraints, and needs for transparency and fairness. Instead of being subsumed by LLMs, evolutionary methods complement them as optimizers of LLM behavior and clean engines below AI-augmented educational platforms.

## 10.3 Importance of optimization in educational contexts

Genetic Algorithms (GAs) and Evolutionary Computation (EC) may provide adaptive solutions for curriculum sequencing, hindrance reduction, dropout prevention, and student engagement scalability. Traditional optimization methods do not manage well the complexity and variability of these processes, whereas GAs and EC are scalable to these challenges and could adapt to changes in educational goals and constraints. For example, some authors (Elshani & Pireva Nuçi, 2021; Hssina & Erritali, 2019; Christudas, Kirubakaran & Thangaiah, 2018; Nogueira et al., 2024) proved that GAs can help sequence curricula in terms of previous experience and complexity level, thus minimizing the potential adverse impacts of cognitive overload and disorientation. This flexibility helps to ensure the creation of bespoke responses that will be sensitive to individual learner outcomes and connected as appropriate with institutional goals, resulting in improvements in experience across the board.

The dynamic nature of educational objectives requires solutions that are adaptable to changes in curriculum materials or in specific student progress. Moreover, traditional ways of doing things are often not flexible or efficient enough for these tasks. As long as the educational systems are concerned, GAs can modify a solution to fit changing goals. Zhou et al. (2011) showed that GAs are suitable in dealing with nonlinear optimization problems, especially when one seeks to reach two or more conflicting objectives at the same time (Deb, 2001; Zhang & Li, 2007).

For instance, multi-objective evolutionary algorithms can be useful for tuning the trade-off between scientific quality and resource efficiency, suggesting their potential in real-world educational scenarios. This trait allows these algorithms to navigate through the demands of institutions with multiple variable situations without compromising the educational output.

GAs and EC are strong contenders among optimization methods and achieve solutions that tackle defining hurdles like local optima, as identified in traditional approaches. With their specialization in handling real-time data, they perform well in quick-change scenarios like e-commerce and recommendation systems. This adaptability is what Chen (2009) recognizes as the power of GAs to develop individualized learning journeys where a unique learning path and pacing are created based on an individual student's progress and needs. Personalizing in such a way prevents learning from becoming stagnant and results in more engaging, fulfilling experiences overall. This ability of GAs to adapt in real-time is the reason that educational methods remain effective as well.

Another area where GAs possess significant potential is promoting and predicting dropout risk in students. These algorithms enable institutions to identify students who most need early alerts and proactive interventions through behavior patterns and engagement metrics. For instance, Xing et al. (2015) and Xue et al. (2016) have shown that predictive models built with GAs are able to outperform some classical classification techniques, finding more accurate dropout indicators, as tested in Queiroga et al. (2020). In theory, these algorithms, as other techniques (Márquez-Vera et al., 2016; Rastrollo-Guerrero et al., 2020), can be supposed to improve retention by predicting and helping students solve challenges early on, thereby preventing them from leading students down a problematic path and towards dropout. These advances suggest the power of GAs and EC in preventing academic failure and promoting educational success.

Operational efficiency is fundamental to the sound management of an institution, and the incorporation of GAs and EC into decision-making outcomes serves as a key driver in improving operations such as course scheduling, or resource allocation that are not always immediately thought as educational problems but, on the contrary represents central questions for institution organizations and the management of time and space in teaching activities and assessments. For example, a mixed-integer genetic and hybrid evolutionary approach can significantly reduce costs and computational time in solving the allocation problem, permitting institutions to adapt quickly to new constraints (e.g., fluctuations in enrollment or resource availability), without considerable loss of efficiency. Customized GAs plus matheuristic functions, namely optimization algorithms using mathematical programming, also offer reduced solution times and can, thus, scale for institutional implementations (e.g., asynchronously in virtual/distance learning). This makes GAs and EC two key techniques that are essential in the current framework for education resources management.

The intricacies of administering resources and directing activities within higher education can require algorithms – multi-objective optimization – to satisfy multiple operational constraints at once. Mixed-integer programming (MIP) based matheuristics can address this issue by combining global search and local feasibility and refinement, integrating the GA exploration and an exact solver. In a heterogeneous university setting defined by comprehensive features, with limited resources for classroom utilization and faculty availability, these hybrid approaches are capable of generating feasible (and near-optimal) timetables. This automates processes not only to make the institution more efficient, but also to provide a more cohesive experience in education for all users (Puchinger & Raidl, 2005; Dunke & Nickel, 2023).

One of the most important targets in educational optimization is actually the personalization of learning pathways, where GAs can be more advantageous. GAs can provide consistency in curriculum sequencing using ontology-based concept maps with the personalization of learning content to meet the needs and preferences of individual learners, as described in Chen (2009). In addition, ontology-driven approaches, considering the difficulty level of course material, are meant to alleviate the problem of cognitive overload that is known to be present in traditional non-adaptive learning settings. These optimized pathways provide students with the ability to reach their educational outcomes faster and more efficiently, leading to better educational outcomes for institutions.

Beneficial for this purpose is its ability to target numerous educational objectives simultaneously, which is one of the great strengths of GAs. For example, GAs could optimize learning routes by considering real-time metrics for adjustments in curriculum. These multi-objective optimization frameworks, made available by EC as well, are essential to balance conflicting goals, such as maximizing engagement and minimizing resource cost (Zhou et al., 2011). The work of Zhou et al. (2011) stresses the benefit of decomposition-based techniques in the context of dividing complex multi-objective problems into feasible sub-problems, as their methods are practical and directly applicable. As a result of this flexibility, Multi-objective Evolutionary Algorithms (MOEAs) are applicable to multiple objectives represented by educational institutions. They effectively tackle a number of constraint-rich problems using methods like memetic algorithms, which combine the capabilities of global and local optimization.

Overall, using GAs and EC for optimization is a promising technique for designing educational systems to fit the intricacies of current demands. Models such as curriculum sequencing, resource allocation, early risk detection, and adaptive systems may assist solutions to be adopted quickly and efficiently because they are adaptable or efficient. Customized GA/matheuristic functions and hybrid solutions also offer reduced solution times and can, thus, scale for institutional implementations, as demonstrated in mixed-integer programming contexts (Yokota, Gen & Li, 1996).

# Chapter 11

## Applications in Education

All the conditions are in place for modern education to increasingly rely on data-informed optimization to provide personalization at scale, respond efficiently when resources are scarce, and intervene early for learners at risk. Genetic Algorithms (GAs) and, more generally, Evolutionary Computation (EC), are very suitable for these situations: they generate a population of candidate solutions which are iteratively improved via variation (crossover, mutation) under explicit objectives and constraints. The result is a strong optimizer that can be applied to combinatorial, non-differentiable, and multi-objective problems that commonly arise in education: curriculum sequencing, test assembly, timetabling, adaptive tutoring policies, and early-warning pipelines.

An extensive overview of Genetic Algorithms in Educational Research in the period from 2020 to 2024 has been published by the authors (De Santis & Minerva, 2026).

In parallel, Large Language Models (LLMs) have become fixtures in educational technology: drafting rubrics and feedback, generating formative questions, supporting tutoring dialogs, and accelerating analytics work (Kasneci et al., 2023). LLMs are trained with gradient methods, but their real-world use exposes discrete design spaces whose performance is observable but not differentiable. This is precisely where EC complements LLMs. The latter can be used for representation, synthesis, and proposal; EC represents the outer-loop optimizer that searches these discrete spaces, reveals tradeoffs, and enforces constraints. In other words: LLMs broaden what we can express; EC formalizes how we decide.

In this chapter, we describe EC applications in education and present how EC, coupled with LLM-assisted workflows, performs personalization, predictive modeling, and other operations in Learning Analytics (LA) and Educational Data Mining (EDM), respecting scalability and governance (fairness, privacy, transparency). To facilitate evaluation, we focus on multi-objective evolutionary algorithms as MOEAs and NSGA-II, that return Pareto fronts rather than single-point solutions, and allow integration of human-in-the-loop choices with respect to local values (e.g., learning gain versus staff effort). In Table 11.1 an Application-Domain at a glance is shown.

**Table 11.1** - Educational optimization domains

Decision variables, objectives, constraints, and metrics across adaptive learning, curriculum, prediction, and tutoring systems and serves as a taxonomy of GA applications in education.

| Application domain | Decision variables | Primary objectives | Hard constraints | Evaluation metrics |
|---|---|---|---|---|
| Adaptive learning (11.1) | Module sequence; remediation inserts; hinting policy | Maximize learning gain; minimize time-on-task; reduce disparity | Prerequisites; workload caps; content exposure limits | Mastery delta; persistence; disparity metrics |
| Curriculum & resources (11.2) | Room/time assignments; instructor loads; sectioning | Minimize conflicts; maximize utilization; improve slot equity | Capacities; availability; accreditation rules | Clash rate; utilization; equity indices |
| Predictive models (11.3) | Feature subset; model hyperparameters; alert thresholds | Maximize early recall; minimize false-positive burden; ensure calibration | Privacy; capacity limits for advisors | AUC/PR; lead-time; calibration; subgroup diagnostics |
| Intelligent tutoring (11.4) | Problem selection; feedback granularity; prompt templates | Maximize mastery; minimize cognitive load; respect latency/cost | Prerequisites; content safety; latency budgets | Learning gains; rubric agreement; abandonment rates |

## 11.1 Adaptive Learning Systems

Adaptive learning systems have the transformative power to personalize instruction and, in real time, adapt their responses to individual learners' changing needs. We will focus on personalized learning pathways based on observed performance and dynamic recalibration of instructional content to maintain engagement and retention. A common thread across the examples is applying computational techniques – especially evolutionary search and optimization – in ways that make personalization lower-cost at scale and inclusive of diverse cohorts. These approaches fit into contemporary analytics pipelines and into LLM-enabled workflows that propose, explain, or prototype alternatives, while evolutionary algorithms perform outer-loop optimization under explicit educational objectives and constraints (see Figure 11.1).

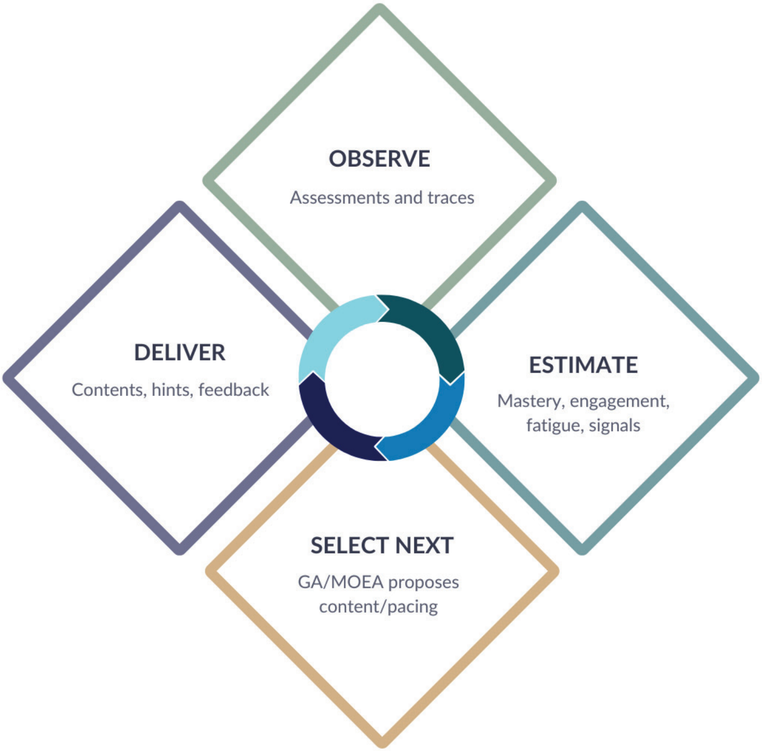


**Figure 11.1** - Adaptive Learning personalization loop

Closed-loop adaptive learning cycle: observe signals → estimate latent states → select next activity with GA/MOEA → deliver and log responses.

### 11.1.1 Personalization of learning paths

Personalizing learning paths using Genetic Algorithms (GAs) is an example of how evolutionary search can facilitate individualized instruction without hand-tuning every detail. In GA-based curriculum planners, a module is encoded as a "gene" and candidate paths are evaluated using fitness functions that assess satisfaction of prerequisites, expected learning gains, and logistical constraints (e.g., the availability of lab facilities or assessment calendars). A recent paper (El Yessefi et al., 2024) describes how such fitness functions can be further adapted to the structures of academic programs, in which skills learned in one term dissipate coherently into the next, relieving teachers of planning pressure while better ensuring curricular coherence.

Beyond static planning, personalization hinges on responsiveness. The approach of Latent knowledge estimation – a mainstay of educational data mining (Baker & Inventado, 2014) – supplies the real-time signals (e.g., mastery probabilities, slip/guess tendencies) that allow the GA to re-weight/resequence a learner's path as evidence is being collected. In practice, this means that a pathway once optimized to fit the average of a large cohort can be modified according to an individual's pace and error patterns: if a student rushes through algebraic manipulation but hesitates on word-problem translation, the search could shift in favor of targeted practice before proceeding.

Multi-objective evolutionary algorithms (MOEAs) also matter here, because personalization rarely optimizes a single goal. NSGA-II (Deb et al. 2002) maintains a diverse set of non-dominated pathways that can balance predicted learning gain, time-on-task, and instructional cost so that instructors or administrators can compare equally defensible options rather than impose an opaque optimal alternative (see, for example, Nogueira et al. 2024).

Adaptive GA planners inevitably raise questions about bias (e.g., when historical data shapes fitness). Mitigations include (i) fairness-aware objectives (e.g., parity of access or probability of on-time completion across demographic groups), (ii) constraints that guarantee exposure to advanced content given sufficient mastery, and (iii) periodic human review of Pareto-front solutions. These are not "nice-to-haves" but design requirements when adaptive paths affect progression or credentialing.

LLM integration can help upstream, at the level of drafting candidate module descriptions, stapled prerequisite links, or formative prompts; then evolutionary search evaluates against explicit metrics (mastery gain per minute, dropout risk reduction) and evolves new proposals. Recent works on evolutionary prompt optimization (e.g. EvoPrompt – Guo et al., 2024; OPRO – Yang et al., 2024, Promptbreeder – Fernando et al., 2023) indicate that downstream task performance can be enhanced via mutation and recombination in the space of prompts/policies, which is applicable for fine-tuning explanation styles (or Socratic questions or language conveying rubric properties) in AI-supported tutoring.

### 11.1.2 Dynamic adjustment based on performance metrics

GAs in adaptive systems shine when they can provide a feedback at the end of the loop: performance signals update fitness, and the algorithm responds by proposing interpretable adjustments to the learner's path. Developers often use binary or mixed encodings to set strict requirements, like prerequisites, and apply variation operators to look for substitutes, reinforcements, or remediations. In adaptive e-Learning, GA pipelines can help reduce the gap between a learner's changing profile and learning goals by choosing the most valuable next steps. Usually, parameterizations with moderate crossover and low mutation rates are used, but these settings need to be adjusted for each problem based on validation groups, rather than simply copying what worked before.

This dynamic adjustment in large-scale educational environments must grapple with volume, heterogeneity, and concept drift. Here, multi-objective search is especially well-suited: NSGA-II, for instance, can balance engagement persistence, mastery growth, and instructor workload, yielding policy sets that generalize across sessions and cohorts instead of overfitting to a single class. The value of a diverse Pareto set is two-fold: it hedges against drift and enables different programs (or campuses) to adopt policies that reflect local capacity and values.

In this context, LLMs can condense formative feedback or generate multiple-choice distractors based on observed mistakes. Evolutionary search then serves as the outer loop of selection and evolution on these artifacts based on later learner responses, analogous to what evolutionary prompt search does on standard NLP (Natural Language Processing) benchmarks (Guo et al., 2024; Yang et al., 2024).

### 11.1.3 Fairness-aware multi-objective adaptive sequencing and governance

As a result of personalization of learning path, adaptive learning systems cannot optimize a single goal. In practice, instructors and administrators want to raise mastery and continuity, while keeping time-on-task reasonable, workloads sustainable, and outcomes equitable across learner subgroups. Framed this way, personalization is a multi-objective problem rather than a single-metric race to the top. Modern evolutionary approaches, especially multi-objective evolutionary algorithms (MOEAs) such as NSGA-II and decomposition-based strategies like MOEA/D (Deb et al., 2002; Zhang & Li, 2007), are well-suited to this reality: they maintain a Pareto front of non-dominated policies – alternative content sequences, pacing rules, or hinting strategies – that trade off learning gain, effort, cost, and equity without forcing premature aggregation into a single weighted objective. This approach is methodologically appropriate and mirrors how educational decisions are (or should) actually made, by comparing defensible alternatives after tradeoffs have been clearly revealed. In education-facing deployments, NSGA-II remains a pragmatic default because of elitism, diversity preservation, and stable behavior, while MOEA/D's decomposition helps when researchers and practitioners need to coordinate multiple objectives across modules, cohorts, and terms.

A central reason (especially in education) to treat adaptive sequencing as multi-objective is fairness. Even when a system improves average outcomes, it can unintentionally widen gaps if recommendations work better for some groups than others. The fairness literature distinguishes group fairness criteria (e.g., equal true positive rates across protected groups) and individual fairness notions (similar learners should receive similar treatment). In student-success settings, group criteria translate into requirements such as parity of "benefit" (e.g., hitting a target mastery threshold) across demographic groups at comparable preparation levels, while individual criteria support consistency of recommendations for near-neighbors in the feature space. Importantly, fairness constraints often conflict with pure accuracy or efficiency, which is precisely why MOEAs are valuable: they expose feasible trade-offs transparently instead of burying them in opaque weights. Classic formulations such as equal opportunity/equalized odds and related constrained-optimization recipes can be incorporated directly as objectives or constraints inside the evolutionary loop (Hardt, Price & Srebro, 2016; Zafar et al., 2017; Mehrabi et al., 2021). In Table 11.2, we show a summary of fairness-aware objectives and constraints.

From an engineering perspective, there are three recurring design patterns. First, a constrained-MOEA formulation treats fairness as a hard constraint (e.g., bounds the disparity in pass-rate improvements between groups) and searches only the feasible region; penalty or repair operators are used when a child solution violates the bound. Second, an explicit multi-objective formulation elevates fairness to a peer objective (e.g., minimize disparity while maximizing predicted learning gain and minimizing total effort), letting the algorithm present a front of policies that span the accuracy-equity spectrum for review. Third, a post-optimization calibration layer tunes operating thresholds per subgroup (e.g., alert thresholds, mastery cut-scores, or intervention intensities) to satisfy fairness constraints while keeping the base sequence fixed; a short GA over group-specific thresholds efficiently finds feasible settings when closed-form solutions are unavailable.

Because adaptive systems run continuously, fairness must be treated as a process rather than a one-off constraint. Performance and parity can degrade due to data drift – enrollment, preparation, or platform behavior changes. In practice, the evolutionary controller should be re-optimized on a cadence tied to the academic calendar (e.g., post add/drop; mid-term) and whenever monitoring detects statistical shifts. This is well-suited for MOEA workloads: fitness evaluations are “embarrassingly” parallel, and cached components (e.g., item difficulties, content costs) make recomputation

**Table 11.2** - Fairness-aware objectives and constraints

Operational fairness metrics (equal opportunity, workload equity, access parity) with definitions and monitoring strategies. It demonstrates integration of ethics within MOEA framework.

| Metric / constraint | Definition (as used) | Where applied | Monitoring |
|---|---|---|---|
| Equal opportunity | Approximate equality of true-positive (benefit) rates across groups | Early-warning thresholds; adaptive promotion | Weekly subgroup TPR/ threshold plots |
| Disparity bound | Upper bound on gap of mastery improvement between groups | Adaptive sequencing policies | Pareto front annotated by disparity |
| Workload equity | Variance cap on advisor caseload / instructor slots | Intervention queues; timetabling | Load histograms; variance trend |
| Access parity | Time-zone or slot-quality parity across cohorts | Synchronous distance schedules | Equity index; heatmaps |

efficient. When evaluation is expensive, surrogate modeling (e.g., simple regressors for mastery gain or engagement response, trained on previous runs) can help accelerate inner-loop fitness estimation, allowing ground-truth evaluations to be reserved for top candidates.

With LLMs, we have helpful outer-loop leverage here, not as replacements for optimization but as instruments to tighten the modeling loop. In curriculum sequencing, for example, instructors can state fairness and feasibility constraints in natural language ("ensure recommended remedial modules do not lengthen time-to-competency disproportionately for older part-time learners") which an LLM could translate into formal predicates and penalty functions that the MOEA can use: an increasingly practical pattern as LLM-based optimizers learn to propose, mutate, and select discrete artifacts (prompts, rules, thresholds) in tandem with evolutionary search. Conversely, evolutionary prompt and policy optimization can be layered around LLM tutors: an EA mutates and recombines prompt templates for Socratic questioning or misconception repair and selects the variants that improve rubric agreement, mastery gain, or parity of benefit across groups. Evolutionary prompt search (e.g., EvoPrompt; Promptbreeder) and LLM-as-optimizer approaches (e.g., OPRO) outperform handcrafted prompts across diverse tasks, which makes them natural building blocks for fairness-aware educational agents that must operate under multiple objectives (Guo et al., 2024; Fernando et al., 2023; Yang et al., 2024).

Evaluation and reporting are as important as optimization. A fairness-aware adaptive system should publish the Pareto front relevant to the decision (e.g., three-objective surfaces spanning learning gain, effort, and disparity), annotate selected points with justifications in plain language, and provide an "optimization card" logging the objectives/constraints in a human-readable form. This helps keep evolutionary search accountable to institutional values rather than purely technical metrics and, at the same time, is congruent with established guidance in Learning Analytics on the principles of transparency, consent, and the obligation to act (Ferguson, 2012; Slade & Prinsloo, 2013; Ungerer & Slade, 2022).

Finally, care is needed with data and measurement. Fairness goals are only as sound as the features and labels on which they rest; bias in historical data can propagate through both predictive models and the adaptive controller (Mehrabi et al., 2021). Practical safeguards include using evolutionary feature selection to minimize reliance on sensitive proxies, running counterfactual or stratified evaluations to detect subgroup failures, and reserving institutional capacity for equitable interventions (e.g., ensuring that sequences recommended for underrepresented learners are actually deliverable given staffing and scheduling). When evidence shows that objectives are incompatible at acceptable levels – a common occurrence in the fairness literature – the Pareto picture allows governance bodies to choose knowingly among the trade-offs rather than assuming a unique optimal policy exists.

### 11.1.4 LLM-augmented adaptivity: evolutionary outer loops

Large Language Models (LLMs) are exceptionally good at generating high-quality alternatives: explanations, worked examples, formative questions, rubric text, and hint sequences. What they do not supply on their own is decision-quality optimization over those alternatives given institutional and pedagogical constraints. In adaptive learning systems, this missing piece is the outer loop: a principled search procedure that selects, refines, and governs LLM-generated candidates with respect to explicit, auditable objectives such as learning gain, rubric agreement, equity, cost, and latency. Evolutionary algorithms (EAs) provide just this capability. They maintain a diverse population of candidates, iterate via variation (mutation, recombination) and selection, and can expose Pareto fronts when objectives conflict, allowing educators to decide among defensible trade-offs instead of settling on an opaque configuration.

We can distinguish two integration levels between LLM and EA.

*EA-driven prompt and policy search:* in LLM-centric tutors and graders, the "genome" can be a prompt template, a tool-use program, or a rubric specification. Population-based prompt optimizers mutate and recombine such instructions, e.g., "show a counterexample before stating the rule", "ask a bridging question that connects prior errors to the new hint". Then they select candidates that maximize downstream metrics measured on held-out tasks or live A/B tests. There is evidence from recent work that evolutionary prompt optimization significantly outperforms the best hand-crafted prompts on a variety of benchmarks, resulting in better explanations, more reliable rubric adherence, and higher student satisfaction in tutoring scenarios (Guo et al., 2024; see also Zhou et al., 2023, for non-evolutionary automatic prompt engineering baselines). In educational deployments, fitness is often multi-objective – e.g., maximize rubric agreement and learning gain while minimizing token cost and response latency – with NSGA-II providing a robust default for discovering well-spread trade-offs (Deb et al., 2002).

*LLMs inside the optimizer loop:* complementary work shows that the model itself can generate and critique candidate solutions in an optimization loop. "Optimization by PROmpting" (OPRO) treats an LLM as a mutation/selection engine that proposes revised prompts, scores them with natural-language rationales, and iterates; this enables natural-language constraint specification (e.g., "avoid overlapping labs for Year-2"; "cap nightly workload at 90 minutes") while retaining EA-style exploration to escape brittle local optima (Yang et al., 2024). More broadly, "LLMs as evolutionary optimizers" demonstrates that LLMs can play the role of operators in combinatorial search without model retraining, useful when institutions must optimize timetables, item sets, or intervention policies around a fixed model (Liu et al., 2024). In practice, this yields co-optimization: LLMs propose and explain alternatives; the evolutionary outer loop enforces constraints and optimizes toward pedagogical goals (see Table 11.3 for a summary on possible LLM-EC integrations).

**Table 11.3** - LLM-EC integration points

It details how evolutionary algorithms interact with LLM components (prompts, hints, rubrics, and routing programs) specifying roles, metrics, and safeguards.

| Component | EA role | Primary metrics | Risks & mitigations |
|---|---|---|---|
| Prompt templates | Mutate/recombine; select via MOEA | Rubric agreement; latency; cost | Safety filters; audit logs |
| Hint policies | Evolve granularity/ordering | Mastery delta; abandonment | A/B with holdouts; teacher review |
| Rubrics | Evolve phrasing/examples | Inter-rater agreement | Calibration; versioning |
| Routing programs | Search tool-use strategies | Task success; latency | Timeouts; fallbacks |

A concrete outer-loop workflow in an adaptive lesson generator can follow these steps:

1. *Generation (LLM):* produce a slate of candidate hints/explanations conditioned based on the learner's recent trajectory and misconceptions, potentially with retrieval-augmented grounding.
2. *Evaluation (analytics):* score each candidate on offline rubrics (coverage, readability, correctness), online signals (click-through, dwell, hint-abandon), and causal uplift proxies (e.g., short-horizon mastery delta).
3. *Selection & variation (EA)*: keep non-dominated candidates, mutate structure (ordering, scaffolds, examples) and wording (reading level, tone), recombine successful fragments (building blocks), and repeat under multi-objective constraints-learning gain, equity (gap reduction), latency, and token cost.
4. *Governance:* log objectives/constraints and publish the current Pareto set for teacher review; guard with safety filters and domain-specific content checks.

To make outer loops decision-grade, instrument them with (i) stratified A/B (or bandit) experiments aligned to ethics policies; (ii) fairness-aware objectives (e.g., minimize disparity in mastery gain across demographic strata alongside global improvement); and (iii) reproducibility safeguards (frozen evaluation sets, seed control, drift monitors). Evolutionary search is naturally parallelizable – candidates can be evaluated concurrently across cohorts – so it fits the operational cadence of large programs and MOOCs. When evaluation is costly (e.g., human rubric scoring), use surrogates (learned predictors of rubric agreement) and caching of fitness components to keep the loop responsive, then periodically re-calibrate surrogates against ground truth.

The EA-plus-LLM pattern adds value especially when: (a) objectives are non-differentiable (rubric agreement, teacher workload, budget), (b) the design space is discrete/structural (prompt templates, item blueprints, schedule policies), and (c) leaders need transparent trade-offs (e.g., +0.07 mastery gain at +15% cost and +0.3s latency). In such cases, LLMs broaden the high-quality design space (many plausible alternatives); EAs structure the search so adaptivity stays aligned with explicit, auditable goals (Deb et al., 2002; Guo et al., 2024; Yang et al., 2024; Fernando et al., 2023).

## 11.2 Curriculum and resource optimization

Curriculum and resource optimization sit at the operational core of education systems: they determine what is taught (and in what order), when and where it happens, and how scarce assets (rooms, instructors, labs, assessment slots) are allocated under hard institutional constraints and softer pedagogical preferences. In practice, these problems are high-dimensional, combinatorial, and multi-objective: a feasible plan must respect prerequisite graphs and credit loads while trading off learning gain against time-to-competency in curriculum design; and in timetabling it must satisfy room capacities, instructor availability, and program clashes while balancing fairness, compactness, and travel or time-of-day equity. Classic surveys and recent syntheses in automated timetabling document the NP-hard nature of these tasks and the need for metaheuristics and hybrids that scale to real institutions (Babaei, Karimpour & Hadidi, 2015; Bashab et al., 2023; Siew et al., 2024; Abdipoor et al., 2023).

Evolutionary Computation (EC) – including Genetic Algorithms (GAs) and multi-objective variants such as NSGA-II – has become a practical default for these settings because it natively handles mixed discrete/continuous design variables, nonlinear penalties, and multiple objectives. Population-based search explores diverse schedule or pathway encodings (e.g., permutations for course/exam placement; integer assignments for rooms and times) while feasibility is enforced via repairs or penalty functions; multi-objective engines maintain Pareto fronts that expose the trade-space administrators actually face (e.g., utilization vs. equity), rather than hiding it inside a single weighted sum. When exact methods are useful locally, matheuristics combine EC's global exploration with mixed-integer programming for polishing, yielding strong results on curriculum-based course timetabling and related allocation problems (e.g., Dunke & Nickel, 2023).

Curriculum optimization adds a pedagogical layer: sequences must respect prerequisite ontologies and cognitive load while adapting to heterogeneity in learner preparation. Here, EC can encode candidate study plans as ordered module lists and score them with composite fitness functions that balance blueprint adherence, predicted mastery gain, and time-to-competency; the same mechanics support dynamic re-optimization as evidence accumulates during a term. Empirical work on ontology-guided sequencing illustrates how such objective formulations reduce disorientation and improve coherence-benefits that carry over when the same ideas are applied at program scale, e.g., multi-semester progression plans (see Chen et al., 2009).

Large Language Models (LLMs) now complement this toolkit at the problem-formulation and rapid-iteration layers. Instructors and staff can express soft constraints, preferences, and “what-if” scenarios in natural language; LLM-assisted pipelines translate these into candidate blueprints or constraint sketches that EC can vet and optimize. Conversely, evolutionary prompt-search frameworks and “optimization-by-prompting” (OPRO, Yang et al., 2024) let institutions treat rubrics, scheduling rules, or content-generation prompts as discrete design variables, evolving them against measurable objectives such as fairness, satisfaction, or downstream schedule feasibility. In short, LLMs broaden the space of plausible alternatives; EC supplies the auditable outer-loop optimization that selects among them under explicit constraints.

Methodologically, this section treats curriculum and resource optimization as first-class, multi-objective decision problems. We can make the objectives (e.g., mastery gain, time-to-degree, utilization, equity) and constraints (e.g., capacities, clashes, accreditation rules) explicit; discuss robust encodings and repair operators; compare single- vs. multi-objective solvers; and, where appropriate, use matheuristics to exploit local structure. Throughout, we emphasize reproducibility (standard benchmarks, public instances), transparency (reporting full Pareto sets), and governance (documenting the values embedded in objective weights), building on established timetabling benchmarks and best practices.

### 11.2.1 Efficient course scheduling and timetabling

Figure 11.2 shows a map of common constraints and objectives in curriculum and timetabling problems, which pose a critical challenge in educational institutions. The figure illustrates the need to balance multiple objectives, including minimizing scheduling conflicts, maximizing classroom utilization, and ensuring resource equity. Genetic Algorithms (GAs) have proven effective in addressing these challenges by automating the scheduling process and optimizing it for both institutional and stakeholder needs. The efficacy of GA-driven course scheduling systems in achieving resource allocation metrics has been studied and compared with random or traditional assignment methods (Abdelhalim & El Khayat, 2016; Cheong, Tan & Veeravalli, 2007; Ngo et al., 2021; Hafsa et al., 2025; Dofadar et al., 2022; Ceschia et al., 2023). By incorporating dynamic performance metrics, such as classroom utilization rates and faculty workload distributions, GAs can achieve more holistic solutions that account for both hard and soft constraints, reducing underutilization through data-driven, dynamic reallocation. Traditional approaches fail to consider subjective factors such as student preferences and instructor workload equity, leading to suboptimal scheduling outcomes. Integrating these variables into GA applications can result in more inclusive and balanced timetables, thereby increasing satisfaction among key stakeholders, such as students and faculty (Zhang, 2022; Mahlous & Mahlous, 2023; Muklason et al., 2017).

Recent implementations demonstrate that GA-enabled tools can streamline task assignment and collaborative processes (e.g., grouping, sectioning) and revise allo-

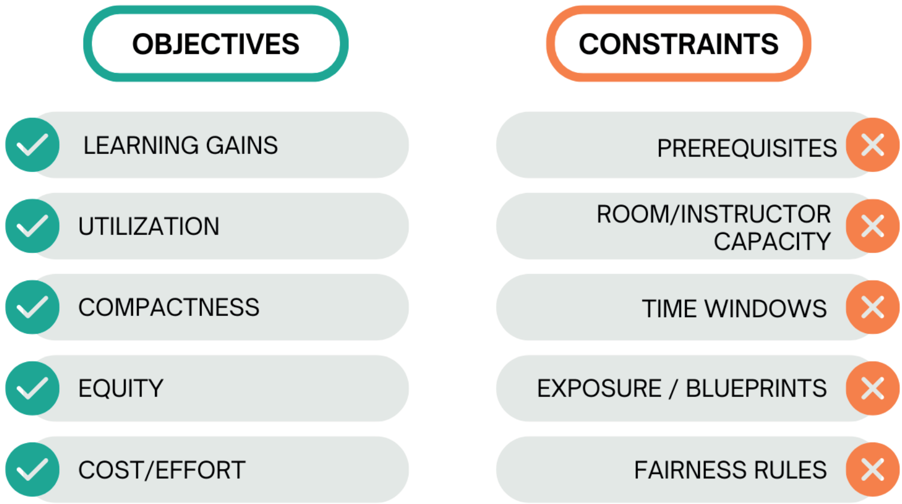


**Figure 11.2** - Constraints vs. Objectives in curriculum and timetabling

Two-column schematic mapping institutional constraints (rooms, instructors, prerequisites) to objectives (utilization, equity, compactness). It aids in problem formulation before encoding for GA search.

cations with minimal human intervention, improving turnaround and stakeholder satisfaction (Li, Ouyang & Chen, 2022; Moreno, Ovalle & Vicari, 2012).

Multi-objective optimization is a strength of Genetic Algorithms for such tasks (Lei, Shi & Yan, 2018; Hafsa et al., 2025). Institutions typically face conflicting priorities: utilization vs. fairness, compactness vs. instructor preferences, and even equity across cohorts. Unlike single-objective algorithms that collapse all optimization problems into a weighted sum, multi-objective evolutionary algorithms (MOEAs), such as NSGA-II, maintain a Pareto set of non-dominated schedules so that trade-offs are visible to and auditable by decision-makers.

In university timetabling (see Table 11.4), GAs' intrinsic flexibility accommodates many multi-dimensional constraints (such as the availability of faculty, sizes of student groups, capacities of rooms, and relations among courses) by encoding schedules, satisfying hard constraints through repair/feasibility mechanisms, and optimizing soft-constraint penalties (like compactness, fairness, or satisfaction of preferences). This adaptability enables the algorithm to create conflict-free timetables that meet institutional requirements, with a massive reduction in manual effort and administrative load. Integrating GAs with intelligent analytics systems enhances their ability to handle large-scale scheduling problems across universities with diverse program structures and campus locations, ensuring equitable and efficient resource allocation (Zhang, 2022; Dofadar et al., 2022). So, another key advantage of GAs is their ability to adapt dynamically to changing variables such as sudden faculty unavailability, fluctuating enrollments, and emerging course offerings. Variation operators (mutation

**Table 11.4** - Timetabling and Resource Objectives with Soft-Penalty Terms

Summary of common objectives in educational timetabling – conflict minimization, utilization, compactness, and equity – with their operational meanings, associated soft-penalty formulations, and observed or target performance ranges. The table helps practitioners map institutional goals into quantifiable optimization criteria for Genetic Algorithm–based scheduling and resource allocation systems.

| Objective | Operational meaning | Typical soft penalty | Observed/target range |
|---|---|---|---|
| Conflict minimization | Avoid lecturer/room/student clashes | Penalty per clash | Zero in feasible solutions |
| Utilization | Fill rooms/slots without overload | Penalty for under-/over-use | ≥ 85–90% aggregate |
| Compactness | Avoid long idle gaps | Penalty per long gap | Context-specific |
| Equity | Distribute unpopular slots fairly | Penalty for disparity | Monitor across cohorts |

to introduce local variations; crossover to recombine high-quality substructures) let the search explore alternative configurations while preserving feasible building blocks, and iterative fitness evaluation steers successive generations toward improved trade-offs, balancing teaching loads, minimizing room/travel changes for students and instructors, and honoring preference profiles. This enables real-time adjustments when unforeseen constraints occur.

Although these mechanisms improve efficiency, the extensive and sophisticated computational infrastructure required can affect scalability issues within larger institutions, becoming at the same time a limiting factor for smaller educational institutions with fewer resources. The application of advanced techniques, hybrid models with local search or mathematical programming and parallelization, can mitigate these challenges while maintaining the robustness and accuracy of GA-based approaches (Ceschia et al., 2023; Dunke & Nickel, 2023; Babaei, Karimpour & Hadidi, 2015).

*Distributed/parallel implementations*. Alba & Tomassini (2002) and Cantú-Paz (2001) investigated the scalability of GA-based scheduling systems, anticipating their capacity to evolve with educational institutional growth and changes. The solutions based on parallel processing techniques described by the authors improve computational efficiency even as the complexity of institutional operations increases. This adaptability ensures operational resilience and supports long-term institutional goals, such as expanding interdisciplinary programs or managing diverse campus locations.

*Hybridization with local search or mathematical programming*. Advanced EC techniques incorporating hybrid models further extend the capabilities of GAs by integrating real-time metrics like course popularity and resource availability into the optimization process (Jiwatode, Schlecht & Dockhorn, 2024). State-of-the-art implementations frequently adopt memetic (hybrid) designs – GA plus domain-specific local improvement – to hasten convergence or improve solution quality on benchmark datasets.

### 11.2.2 Allocation of resources in digital environments

Optimization of the allocation of learning resources – academic spaces, teacher calendars, and digital tools – is one key problem that Genetic Algorithms (GAs) address effectively (Kučak, Juričić & Đambić, 2019; De Santis & Minerva, 2026). Starting from fitness evaluation mechanisms, GAs systematically assess and generate high-quality solutions in terms of minimizing scheduling conflicts and maximizing the utilization of available resources, all while respecting institutional constraints and priorities (Mitchell, 1995; He, 2022; Qiu, 2022). Unlike fixed heuristics, GAs evolutionary operators (crossover and mutation) are capable of injecting controlled variation and exploring diverse scheduling configurations, enabling consistent improvement across generations.

Several studies (Hurmuzi et al., 2023; Zandavi, Chung & Anaissi, 2021; Christudas, Kirubakaran & Thangaiah, 2018; Chen et al., 2009) underlined the contribution of GA-enabled systems in virtual and distance learning, where traditional methods often tend to understimate the unique challenges of blended education. In fact, online and blended delivery modes introduce further constraints: synchronous vs. asynchronous activities, tutoring, bandwidth, and proctoring windows with shared hybrid spaces (Hurmuzi et al., 2023; Zandavi, Chung & Anaissi, 2021; Tan et al., 2017; Minerva et al., 2024a). By optimizing room allocation and virtual resource distribution, GAs can ensure the equitable accessibility of high-demand resources, even in resource-constrained environments. Moreover, GA relevance in online and blended settings can also be seen in their ability to dynamically adapt (for instance, to fluctuations in enrollment trends or institutional priorities) and to facilitate inclusivity by considering demographic diversity, student preferences during the scheduling process, and equity in resource allocation. GAs adapt well to the highly flexible nature of virtual learning contexts, although their dependency on real-time data analytics in optimization can also lead to complications involving privacy concerns, equity among groups, and a heavy computational load, which calls for the exploration of more secure and efficient frameworks.

The integration of GAs with real-time Learning Analytics also benefits virtual environments by supporting personalized allocation of digital resources, for instance, for proctoring exams, remote lab availability, or tutoring hours. In online higher education, decision trees and other learners can be wrapped by GAs for feature selection and threshold tuning (Xue et al., 2016; Alias et al., 2024), thereby increasing the precision of recommendations, such as nudging students towards synchronous support.

In resource-constrained contexts – especially at scale and at a distance – EC techniques provide measurable advantages over manual ad-hoc allocation. For example, early-warning pipelines that identify at-risk students can be combined with GA-driven allocation rules so that scarce tutoring hours or advising appointments are prioritized where they are most likely to reduce dropout; studies in virtual universities report strong predictive performance, enabling earlier, more targeted support (Hao, Gan &

Zhun, 2022). As virtual enrollments grow (Minerva et al., 2024b), the challenge becomes sustaining predictive fidelity and allocation responsiveness without prohibitive compute costs. Surrogate models, caching, and parallel evaluation can intervene and keep GA runtimes manageable.

## 11.3 Predictive models for student success

Predictive modeling has become a useful tool for identifying at-risk students early so to provide timely, appropriate support. This attitude responds to the purposes of educational institutions focused on learner outcomes and equity. In Learning Analytics and Educational Data Mining (Ferguson, 2012; Papamitsiou & Economides, 2014; Lang et al., 2022; De Santis, 2022), these models aggregate multimodal traces – enrollment and demographic information, activity and engagement logs, assessment records, discourse signals, and advising notes – to estimate student risk levels and, as a consequence, guide support actions. The aim is not solely to achieve high predictive accuracy, but to facilitate effective and data-driven decision-making. In practice, this means showing risks far enough in advance to take action and leveraging limited advising resources only where they'll be most impactful. The process must document trade-offs among sensitivity, precision, lead time, and fairness to boost transparency and allow academic governance to make informed and accountable decisions (Rastrollo-Guerrero, Gómez-Pulido & Durán-Domínguez, 2020; Slade & Prinsloo, 2013).

Methodologically, contemporary pipelines combine robust feature engineering with model selection and calibration tailored to the institutional context (see Table 11.5). Evolutionary Computation (EC) contributes in two key ways. First, it supports wrapper-based feature selection and hyperparameter tuning, which improve generalization and interpretability (see Xue et al., 2016, for a state-of-the-art work and Queiroga et al., 2020, for an example paired with ML models). Second, it enables multi-objective optimization (e.g., NSGA-II), so institutions can balance competing priorities such as early detection of at-risk students and the workload of advisors. These approaches are particularly effective in online and blended learning environments, where data are high-dimensional, imbalanced, and subject to drift (Márquez-Vera et al., 2016; Albreiki et al., 2021; Waheed et al., 2020). In such settings, evolutionary pipelines often outperform static approaches as they better adapt to new conditions associated with early dropout and failure prediction.

However, deploying these systems requires rigor and consideration that go beyond established metrics. Using temporal cross-validation and cohort-out evaluation helps avoid information leakage and concept drift. Probability calibration supports risk thresholds that align with institutional capacity; cost-sensitive objectives reflect the asymmetric harms of missed interventions versus false alarms (Papamitsiou & Economides, 2014; Rastrollo-Guerrero, Gómez-Pulido & Durán-Domínguez, 2020). Fairness is not an afterthought but a first-order consideration: models should be stress-

**Table 11.5** - Key Components of Early-Warning Pipelines

The table outlines the core elements of GA- and EC-enhanced early-warning systems for student success. Each component – feature engineering, model selection, threshold tuning, and triage – is linked to key design choices and best-practice guidelines. It supports transparent, reproducible setup of predictive models that balance accuracy, interpretability, and operational feasibility in educational analytics.

| Component | GA/EC role | Key choices | Good-practice notes |
|---|---|---|---|
| Feature engineering | Wrapper selection over candidate indicators | Attendance volatility; quiz latency; forum centrality | Temporal CV; guard against leakage |
| Model selection | Hyperparameter search; compactness | Tree ensembles, linear, or calibrated margins | Report stability & sparsity |
| Threshold tuning | Multi-objective tuning (recall vs workload) | Group-specific thresholds if required | Publish trade-offs; align to capacity |
| Triage & playbooks | Queue ordering under constraints | Risk bands; intervention types | Log outcomes; run small A/Bs |

tested for disparities across student groups, and, where appropriate, optimized with fairness-aware constraints or penalties to mitigate disparate mistreatment while preserving utility (Hardt, Price & Srebro, 2016; Zafar et al., 2017; Slade & Prinsloo, 2013). These practices are essential if predictive services are to support, rather than undermine, institutional commitment to access and inclusion.

Large Language Models (LLMs) extend these systems by converting unstructured data – forum posts, reflective essays, help-desk chats – into structured indicators of student behavior, including confusion, engagement, and help-seeking. Because the design of prompts and extraction policies is discrete and non-differentiable, they benefit from evolutionary outer loops that search over prompts, exemplars, and rubric wording under task-specific objectives (e.g., inter-rater agreement, calibration error), yielding more stable and portable text-derived features (Guo et al., 2024; Yang et al., 2024). In this setup, LLMs generate representations and candidate explanations, while evolutionary algorithms optimize prompts, thresholds, and intervention strategies.

The remainder of this chapter explores these themes in more detail, focusing on early risk detection, intervention design, and governance frameworks that integrate privacy, fairness, and explainability into the modeling lifecycle. The first paragraph presents the basics of developing early warning systems, with a focus on feature selection methods and hybrid solutions, and highlights the roles of privacy, transparency, and fairness in student profiling. The second one describes interventions, such as recommendations or group formation, that can target solutions for learning and improve student retention.

### 11.3.1 Early identification of at-risk students

A popular area of investigation in which GA has shown potential is the early identification of at-risk students. GAs improve predictive models by using feature selection techniques to reduce the dimensionality of academic challenges, such as attendance, time spent on tasks, interaction data, engagement, and study habits into key indicators (Al Farissi, Dahlan & Samsuryadi, 2020; Hao Gan & Zhu, 2022; Sahlaoui et al., 2024). They refine predictive pipelines by pinpointing the most informative variables and discarding redundant ones. This fine-grained feature-selection process leads to greater predictive accuracy and enables institutions to proactively address the specific difficulties and necessities of at-risk students, improving computational efficiency, minimizing false positives, and supporting precisely targeted interventions. However, despite these advantages, in line with the ambition for more equitable and inclusive deployment, opportunities to move towards generality should be explored; specifically, approaches that examine how well models generalize across courses, institutions, modalities, and metrics (e.g., learner demographics).

In feature selection, a relevant strategy is error-correcting output codes (ECOC) integrated with evolutionary search – appearing in the literature as ECOC-GA – that associates coding schemes such as one-versus-one (OVO) and direct-reduction (DR) with a genetic wrapper to improve multi-class risk prediction. ECOC-style pipelines augmented with GA-based selection (Dietterich & Bakiri, 1995; Allwein, Schapire & Singer, 2000; Xue et al., 2016) can be compared with baseline ensembles (e.g., support vector machines, naïve Bayes, logistic regression) across multiple datasets in blended learning contexts, with the evolutionary component improving adaptability under distributional shift (Tan et al., 2017). These results align with the broader ECOC evidence base in multi-class learning (Dietterich & Bakiri, 1995; Allwein, Schapire & Singer, 2000). Adaptability is particularly valuable in large-scale, heterogeneous environments such as MOOCs or virtual universities, where data are extensive and varied. Nevertheless, translating these gains to very large cohorts raises questions about computational cost and latency; future work should prioritize scalable implementations (e.g., parallel fitness evaluation, caching) without eroding accuracy.

The choice and quality of input features remain decisive for real-world usefulness. Xie et al. (2021) show that ECOC–GA pipelines recover variables like attendance volatility, interaction frequency, and assignment scores as high-leverage outcome predictors. In parallel, studies using big data from virtual learning environments (VLEs) show that temporal interaction patterns and assessment trajectories provide actionable signals for early risk detection and for the design of intervention strategies (Waheed et al., 2020; Queiroga et al., 2020). These benefits, however, presuppose reliable and consistent data capture across platforms and courses. Institutions that implement early-warning analytics should consider investing in consistent data-capture infrastructure and quality-assurance processes to minimize missingness, drift, and inconsistent logging practices.

Hybrid approaches that combine GAs with supervised learners (e.g., support vector machines, tree ensembles) have shown better performance in classifying/predicting at-risk students. Sreenath and Jeyakumar (2016) review prior work showing that GA-based feature selection and parameter tuning – such as GA-enhanced SVMs – can improve classification performance in educational data mining contexts. Their survey highlights the flexibility of evolutionary approaches for optimizing model structures and extracting compact rule sets, particularly in personalized learning applications. An innovative approach is to implement multi-objective evolutionary models integrated with decision-tree classifiers – such as the MmGA (modified micro genetic algorithm) combined with a C4.5 decision-tree classifier (Tan et al., 2017) – which identify predictors of student achievement while balancing accuracy and parsimony. This line of work reports substantial feature reduction with similar accuracy. These models are adept at the computationally efficient reduction of educational datasets into compact, high-signal parameter sets that still produce strong performance. However, the long-term applicability of such parsimonious, tree-based models across varied ecosystems warrants continued validation, especially when computing and data engineering resources are limited.

It is clear that profiling and modeling learning behaviors are central components of predictive analytics for identifying at-risk students. Evolutionary Computation (EC) approaches excel at synthesizing multidimensional signals – enrollments, attendance, task completions, engagement – into robust risk profiles that can trigger relevant interventions. At the same time, the power of profiling results in governance obligations: privacy, transparency, and fairness must be built in from the beginning. Ethical analyses in Learning Analytics emphasize equality of opportunity and the need to explicitly report trade-offs. Therefore, once again, we have to underline that EC-based pipelines should incorporate fairness-aware objectives and document Pareto trade-offs in accessible terms (Hardt, Price & Srebro, 2016; Zafar et al., 2017; Ungerer & Slade, 2022).

In conclusion, the integration of GAs and EC has contributed to the development of research and applications on early identification of at-risk students by optimizing feature selection and enhancing predictive accuracy. Models such as ECOC-GA and MmGA combined with C4.5 exemplify the promise of evolutionary search when objectives conflict and data are noisy. There are still open challenges such as scalability, durable generalization, data reliability, and ethical alignment. Nevertheless, the methodological trajectory is clear: evolutionary pipelines, governed by fairness-aware objectives and sound validation practices, can empower institutions with earlier, equitable, and actionable support for students most at risk.

### 11.3.2 Proactive interventions to improve retention

In addition to early identification of at-risk students, proactive interventions are crucial for improving student retention in educational programs. Using a multivariable evaluation strategy, institutions can design customized support systems comprising course recommendations, targeted tutoring, or personalized, structured study plans that allocate resources efficiently and ensure that the most vulnerable students receive timely, high-impact support. GAs can be applied to face these necessities.

Evolutionary Computation (EC) methodologies, more broadly, have proven promising for integrating adaptive strategies into personalized learning systems, with measurable effects on retention. Christudas, Kirubakaran and Thangaiah (2018) experimented EC capacity to adjust content formats based on real-time performance metrics so that students receive tailored resources calibrated to content difficulty and student learning styles. They highlighted the value of combining behavioral and performance signals so that intervention content aligns with a learner's characteristics. Elshani and Pireva Nuçi (2021) demonstrated that GA-based models can optimize personalized learning paths by accounting for content difficulty and conceptual relationships. Models based on GA indeed addressed the interdependencies among the underlying factors of learning-object difficulty, time-on-task, and actual study behaviors, thereby changing how personalized interventions are constructed. This adaptivity has to keep learners in a productive band of difficulty, neither so challenged they feel overwhelmed nor so unchallenged they are bored or disengaged, which reduces frustration and dropout. The same authors reported that retention improved with GA-generated learning paths, reaching values that were even higher than those of the worst path generated by GA, surpassing those of traditional, non-adaptive approaches. As already noted, they also sharpen issues of fairness and inclusion: algorithms should be audited to make sure that they do not inadvertently disadvantage subgroups as a consequence of representation bias in historical data, and objectives or constraints based on fairness should also be included along with accuracy (Hardt, Price & Srebro, 2016; Zafar et al., 2017; Slade & Prinsloo, 2013).

Collaborative learning environments also benefit from EC, particularly in data-informed group formation that promotes peer support. Moreno, Ovalle and Vicari (2012) demonstrated that, considering knowledge levels, communication skills, and leadership tendencies, EC-driven grouping can increase satisfaction and retention relative to heuristic grouping. By balancing intra-group heterogeneity with inter-group comparability, EC supports mutual help dynamics, lowering the likelihood of disengagement. Group composition can also be adjusted over time in response to evolving evidence, ensuring that struggling students are paired with peers who can provide effective support. Still, questions remain about scalability to very large cohorts and about the ethics of algorithmic assignment; clear policies, opt-out paths, and equity audits are important to ensure inclusivity.

The application of GAs to large-scale online environments such as MOOC platforms is particularly notable because these environments produce huge, rapidly shifting datasets (Mir & Khan, 2026; Stracke et al., 2019; Reich & Ruipérez-Valiente, 2019; Ruipérez-Valiente et al., 2022). In the context of MOOCs, GA-based systems can (i) analyze behavioral traces to surface actionable insights identifying students likely to disengage through analysis of activity patterns and performance trajectories; and (ii) reassess progress on an ongoing basis in order to adapt interventions in near-real time. But while this mechanism of adaptivity is a strength, it poses an equity challenge as well: systems that rely on substantial computing power and continuous data may be harder to use at institutions with fewer technical resources. This underscores the need for resource-efficient variants and robust governance for continuous monitoring, including protections for learner privacy and strategies to minimize bias.

Large Language Models (LLMs) can expand the space of candidate intervention messages (e.g., feedback styles, nudges, rubrics), while EC provides an outer-loop optimizer that searches over these alternatives under explicit objectives (learning gains, rubric agreement, fairness, staff effort). Population-based prompt search (e.g., EvoPrompt) can mutate and recombine message instructions, selecting those with the best downstream effects in A/B tests (Guo et al., 2024). Complementarily, Optimization-by-Prompting (OPRO) lets an LLM propose and evaluate "mutations", with EC-style selection avoiding brittle local optima (Yang et al., 2024). In retention workflows, this pairing can optimize proactive outreach (e.g., message tone, timing, and specificity) while respecting institutional constraints and ethical safeguards (Slade & Prinsloo, 2013).

For institutions beginning this journey, a phased approach is pragmatic: start with GA-based feature selection wrapped around existing early-warning models (Albreiki, Zaki & Alashwal, 2021; Waheed et al., 2020; Queiroga et al., 2020), add multi-objective policies that explicitly trade off recall and staff workload, and pilot LLM-EC intervention design in limited A/B tests with human oversight. Along the way, document goals, constraints, and Pareto trade-offs in a plain-language accessible way so stakeholders understand the rationale behind any recommended intervention.

In conclusion, proactive interventions driven by GAs and EC offer a transformative pathway to improved retention. These techniques take diverse learners' needs as input while customizing feature selection, adaptive learning paths, and collaborative support to deliver optimal solutions to educational problems. Solving open challenges of scalability, accessibility, and ethics will not only allow us to build effective systems but also ensure they are equitable and sustainable across contexts.

## 11.4 Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) sit at the intersection of computational intelligence and education, delivering individualized instruction through learner models that infer mastery, pedagogical policies that choose what to do next, and interfaces that communicate feedback in real time (Woolf, 2009; Ma et al., 2014). Well-designed ITS can approach the effectiveness of human tutoring and reliably outperform traditional classroom practice on measures of learning and transfer (VanLehn, 2011). Within this architecture, Genetic Algorithms (GAs) and the broader family of Evolutionary Computation (EC) can serve as outer-loop optimizers: they search over problem (selection policies, hinting strategies, and content sequences under multiple, often conflicting objectives) maximizing learning gains while minimizing time-on-task, cognitive load, or instructor effort, yielding transparent trade-off frontiers rather than a single brittle choice.

GAs/EC are particularly well suited to the discrete, constraint-rich design spaces that characterize ITS: ordering and pacing of activities, mastery thresholds, spacing and interleaving schedules, and the calibration of feedback granularity. Population-based search maintains diversity to avoid local optima and can incorporate feasibility preserving operators that respect prerequisites and assessment calendars. Foreshadowing today's adaptive platforms, early lines of research demonstrated how evolutionary methods can even shape the inner learning mechanisms evolving neural topologies and parameters to meet task-specific criteria (Harp, Samad & Guha, 1989).

Large Language Models (LLMs) now extend ITS capabilities by generating fluent, context-aware explanations, Socratic prompts, and rubric-aligned feedback. Yet these generative choices introduce vast, non-differentiable design spaces (prompts, policies, exemplars) that still require principled optimization. Here, EC remains complementary: evolutionary prompt search can iteratively mutate and recombine instructional prompts to improve downstream learning or rubric agreement (Guo et al., 2024; Yang et al., 2024). In combination, LLMs broaden what the tutor can express; EC ensures that expression is aligned with explicit, auditable objectives and institutional constraints.

Lastly, as ITS scale across modalities and populations, governance becomes first-order. Fairness-aware objectives, privacy-preserving data pipelines, and clear documentation of optimization goals and constraints are necessary to ensure that adaptivity remains equitable and trustworthy (Slade & Prinsloo, 2013).

The section that follows examine these themes in depth: how ITS personalize feedback, how EC can intervene in teaching strategies and environments, and how the retrieval of information generated in these systems can be facilitated by visualization and dashboards.

### 11.4.1 Real-time feedback and adaptive teaching strategies

Even though other algorithms are commonly used to develop ITS (ML, NLP/LLM, probabilistic models), a recent multi-agent ITS, EvoLogic, incorporates a Genetic Algorithm for teaching logic. In this system, GAs are used as a search mechanism over the space of logic solutions, providing a set of possible solutions (not just one), and simulating reasoning in natural deduction and propositional logic (Galafassi et al., 2023). The ITS is composed of three agents (pedagogical, specialist, and interface) and a memory structure that stores the best 10% solutions generated by the GA. When a student interacts with the system and composes a message on the resolution of logic problems via the interface agent, the input is processed by the pedagogical agent, which contains the teacher and student models and determines whether to evaluate the correctness of the solution created by the student for known problems or to ask the specialist agent to generate a solution for unknown ones. In this ITS, the teacher does not need to manually code all solutions; instead, the GA determines a range of solutions by applying crossover and mutation mechanisms to diversify them and avoids infeasible steps. The teaching approach adopted is example-based learning, where students are guided through alternative solution paths, and relevant examples are shown, illustrating how rules similar to those they should use are applied or how other procedures have led to the conclusion.

EvoLogic demonstrates that the application of Genetic Algorithms (GAs) in real-time feedback systems represents a further tool for advance in the personalization and adaptability of educational experiences.

In addition to a system that provides real-time feedback, such as EvoLogic, GAs are recognized for dynamically adapting learning content and teaching strategies to individual needs by analyzing learner performance data, such as task-completion times and correctness rates.

Speaking of collaborative teaching methods, Shin-ike and Iima (2008) proposed a hybrid procedure combining GA and neural networks that optimises pedagogical decisions by modelling learner performance to adapt teaching actions to student characteristics. In the paper, the neural network predicts the learning scores of learners paired with others based on learners' cognitive abilities and personality traits (measured using the Synthetic Personality Inventory), while the Genetic Algorithm searches for a near-optimal coupling that maximizes overall learning results. The system provides a structured method for forming learner groups, offering educators a data-informed strategy to improve learning results and the effectiveness of collaborative learning activities through optimized group composition (Shin-ike & Iima, 2008). The integration of neural networks further enhances adaptability: predictive models trained on historical data (e.g., prior accuracy, pace, and even self-report measures) predict short-term learning outcomes, allowing more appropriate, just-in-time interventions (Shin-ike & Iima, 2008). These two components (Gas and NN) create an iterative feedback loop, merging the power of predictive analytics and optimization that result in flexible solutions for various learner needs. By automating much of the drilling down for personalization, those systems also offload the manual workload from educators, providing time for mentoring and academic planning. Instead, Xing, Guo, Petaković, and Goggins (2015), working in the environment Virtual Math Teams with GeoGebra, used interpretable Genetic Programming (GP) to generate predictive models that connect the constructs of activity theory (Subject, Rules, Tools, Division of Labor, Community, Object) to students' performance and participation patterns. By linking these variables to trace data from student interactions in a collaborative environment (e.g., participation patterns and tool use), GP provides interpretable if-then rules that help instructors identify learning gaps and offer targeted support.

At the core of these systems lies the GA fitness-evaluation function, which quantifies the alignment between instructional content (or activities, as in the previous case) and each learner's current needs or choices. For example, a GA can weight recent correctness rates and latency to dynamically recalibrate and refine learning pathways, ensuring students are neither overwhelmed by excessive complexity nor disengaged by overly simplistic material. In the next chapter, cases on curriculum sequencing and recommendation systems – central in ITS and adaptive solutions – will be described in detail (Christudas, Kirubakaran & Thangaiah, 2018; Hssina & Erritali, 2019; Chen, 2009; El Yessefi et al., 2024).

The adoption of visual analytics in EC techniques further complements the use of evolutionary techniques and, as a consequence, in our discussion, adaptive teaching. Visualization tools can summarize information generated during evolutionary processes, providing intuitive insights and effects that are otherwise less easily revealed (Llorà et al., 2006). Moving to the education context (Ifenthaler & Yau, 2020), such visualizations that summarize and organize information can assist with recognizing sub-optimal learning strategies and iteratively improve teaching methodologies. For instance, visual traces of error patterns enable instructors to pinpoint consistent misconceptions within specific topical clusters, motivating targeted micro-remediation and richer feedback. In this regard, moving beyond the evolutionary approach, there is extensive discussion within the field of Learning Analytics concerning dashboards and their role in tracking the progress of a class or a specific student, predicting trends, implementing monitoring and alert systems to prevent dropouts, and to encourage/sustain participation (Schwendimann et al., 2017; Verbert et al., 2013; Jivet et al., 2018; Sedrakyan, Mannens & Verbert, 2019; Masiello et al., 2024; De Santis et al., 2024).

In conclusion, the integration of Genetic Algorithms and Evolutionary Computation into real-time feedback systems, and, more generally, into teaching activity optimization, can constitute a step change in adaptive education. Such approaches improve personalization, multiply solutions, and increase transparency by using procedures that go beyond the black box of some AI tools, adopting visual analytics, and supporting inclusivity by accommodating heterogeneous learners with diverse needs and latent states. Through continuous, evidence-driven improvements in teaching strategies, EC-enabled systems are well-positioned to transform learning experiences across modalities and scales.

# Chapter 12

## Case studies and real-world implementations

In previous chapters, we have described the role that Genetic Algorithms (GAs) and the broader family of Evolutionary Computation (EC) can assume in education: even if they are less used if compared to other techniques in educational research (Charitopoulos, Rangoussi & Koulouriotis, 2020; Ouyang & Chen, 2022), presented real-world deployments make them not abstract curiosities, but optimizers that can improve learning at scale, daily teaching solutions, and equity outcomes. This chapter assembles case studies across four settings: large-scale educational platforms (MOOCs and adaptive systems), personalized learning pathways in online environments, curriculum optimization tools in higher education, and resource allocation (mainly in distance learning). The aim is to show how population-based search translates institutional and educational goals into auditable decisions under real constraints in numerous tasks. In most of the case studies reported, optimization is treated as multi-objective by default, reporting trade-offs rather than a single "winner" solution and using well-established methods (e.g., widely cited NSGA-II and MOEA/D) to surface Pareto fronts when objectives conflict (e.g., completion vs. cost; accuracy vs. fairness; utilization vs. satisfaction) (Deb et al., 2002; Zhang & Li, 2007; Zhou et al., 2011).

Methodologically, the cases revolve around few principles: (i) clear objectives and constraints stated up in advance and mapped to measurable key performance indicators (conflict rates, room utilization, recall of at-risk flags, preferences, latency targets); (ii) rigorous evaluation via stratified cross-validation, online experiments, and stability checks; (iii) human-in-the-loop governance, where instructors and administrators can pin hard constraints, veto candidates, or explore alternatives through interactive visual analytics (Llorà et al., 2006); and (iv) fairness, privacy, and transparency designed in, not bolted on such as optimizing equality-of-opportunity (Hardt, Price & Srebro, 2016) alongside accuracy and documenting the resulting trade-offs for stakeholders (Slade & Prinsloo, 2013; Drachsler & Greller, 2016). In Figure 12.1, we show an example of data flow into EC engines for decision support.

The chapter is organized as follows: section 12.1 examines large-scale platforms, showing how EC stabilizes personalization and operations under cohort drift and de-

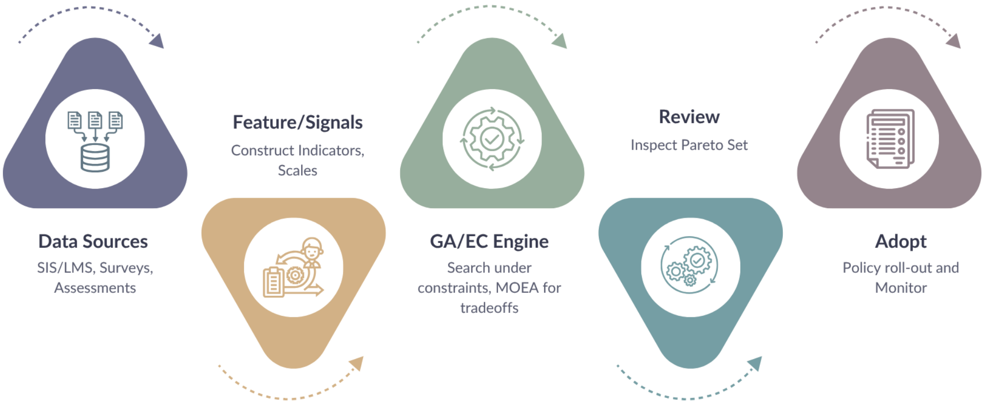


**Figure 12.1** - GA-enabled adaptive learning pipeline

Blueprint for case studies: data sources and signals flowing into EC engines, with review of Pareto sets and adoption into practice.

mand shocks, and how fairness constraints are co-optimized with performance. Section 12.2 focuses on personalized pathways in online learning, where GA/EC engines are used to adapt content and pacing to real-time traces and contribute to the formation of effective work groups among students. Section 12.3 presents curriculum optimization tools used in higher education to design feasible timetables. Section 12.4 closes with case studies, where multi-objective schedulers are used to allocate human, material, and scientific resources together with targeted interventions.

These cases show that if institutions make their goals explicit and let evolutionary search navigate the hard trade-offs, adaptive, equitable, and accountable educational systems become practically achievable.

## 12.1 Large-scale educational platforms

With the continuous evolution of education landscapes, large-scale platforms, especially those developed for delivering Massive Open Online Courses (MOOCs) or institutional formal degree courses, play a prominent role in defining the quality of the learning experience across heterogeneous users.

Built on the analytics foundations established in Learning Analytics and Educational Data Mining (Ferguson, 2012; Lang et al., 2022; Baker & Inventado, 2014; Papamitsiou & Economides, 2014), this paragraph examines how Genetic Algorithms (GAs) and broader Evolutionary Computation (EC) can be embedded within these platforms to optimize at scale. Under hard operational constraints, GAs and EC can face sequencing and personalization of content, real-time orchestration of learning activities, and resource allocation. In high-enrollment settings, heterogeneity and dropout are persistent challenges: evolutionary, multi-objective optimizers (e.g., NSGA-II) provide decision-level trade-offs among completion, engagement, latency, and cost, while respecting constraints on instructors, rooms, or computational budgets (Deb et al., 2002; Reich & Ruipérez-Valiente, 2019; Reich, 2022). To support trustworthy adaptivity at this scale, platforms must instrument reliable telemetry and evaluation: interoperable event streams (e.g., xAPI, Caliper) enable fine-grained measurement; A/B and counterfactual evaluations close the loop from optimization to impact; and governance guardrails address privacy, consent, and fairness in automated interventions (1EdTech Consortium, 2020; Advanced Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; Drachsler & Greller, 2016; Slade & Prinsloo, 2013). Finally, we highlight emerging LLM-augmented workflows based on evolutionary approach, where Large Language Models broaden the design space of explanations, hints, and rubrics, and EC serves as an auditable outer loop that searches those spaces against explicit pedagogical objectives, already demonstrated in optimization-by-prompting and evolutionary prompt search (Guo et al., 2024; Yang et al., 2024).

These applications and threads will be retraced in the sections that follow on MOOCs (Kizilcec et al., 2017; Mir & Khan, 2026) and adaptive systems, with the awareness that much more can be achieved in this sector with the use of GAs. Together with dynamic personalization, we emphasize scalable operations and responsible evaluation in increasingly complex environments. The island-model deployment, as illustrated in the example in Figure 12.2, can more readily provide scalable solutions by allocating distinct evolutionary subpopulations across different cohorts or programs. A core migration policy coordinates cyclic adjunctions of candidate solutions that strike a balance between the exploitation and exploration across the large-scale system.

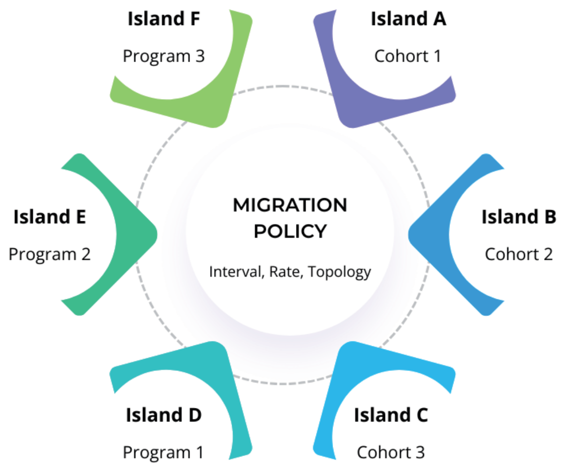


**Figure 12.2** - Island-model architecture for distributed GA deployment

Island-model deployment for large cohorts/programs with migration policy at center to balance exploration and exploitation.

### 12.1.1 MOOCs and adaptive learning systems

As just introduced, Massive Open Online Courses (MOOCs) and adaptive learning systems have been established as actual, scalable, and flexible models of modern education addressed to heterogeneous audiences (Mir & Khan, 2026; Reich & Ruipérez-Valiente, 2019). In real deployments, Genetic Algorithms (GAs) have been used to optimize personalized course pathways and cohort-level decisions rather than acting as purely theoretical constructs. For instance, GA-enabled grouping methods have been tested on higher education courses delivered online and in blended formats to create and "adapt" collaborative teams that balance prior knowledge, engagement habits and preferences; these deployments produced significant improvements in group processes and improve quality, showing how an evolutionary search across groupings can be used to put personalization into effect at scale (Shih et al., 2018; Li, Ouyang & Chen, 2022; Moreno, Ovalle & Vicari, 2012). Likewise, GA-based heuristic solutions for scheduling academic planning ("adapting" module sequences semester by semester while honoring prerequisites and student goals) have been shown to work on end-to-end feasibility out of the labs (El Yessefi et al., 2024; Chen, 2009). We will elaborate on the details of these cases in subsequent sections. In some (probably too few) research, the optimization takes place on live platform data where, using quiz scores, trends in progression, and clicks, it can be achieved that suggested paths remain aligned with learner performance and not static syllabi (Hssina & Erritali, 2019; Ouyang, Zheng & Jiao, 2022). By improving the relevance and timing of content, these implementations directly target high dropout rates common in large online cohorts, translating classic evolutionary search principles into operational improvements (Goldberg, 1989). The scalability of evolutionary approaches is particularly germane to MOOCs, where platforms must accommodate thousands of concurrent learners.

Production analytics pipelines are increasingly blending evolutionary search with at-scale telemetry to determine when to branch, repeat, or enrich content. For example, real-world video learning analytics using interaction traces such as pause, rewind, and replay (Hasan et al., 2020) could identify hard segments of videos to target support for students. While these systems combine multiple machine-learning models, evolution components are used to search over candidate intervention policies (what to show when, and to whom) within platform constraints, enabling course teams to iteratively converge on the types of policies that provide robust learner outcomes.

Beyond instructional sequencing, Evolutionary Computation (EC) has been applied to operational aspects that matter in large online programs and MOOC-for-credit offerings, namely, instructor assignment, synchronous session planning, and proctoring/resource allocation. University deployments of GA-based timetabling, originally designed for campus scheduling, have been adapted to hybrid and online contexts to reconcile instructor availability, cohort time zones, and virtual room capacities, reducing conflicts and increasing utilization in high-demand courses (Abdelhalim & El Khayat, 2016; Ceschia, Di Gaspero & Schaerf, 2023; Hurmuzi et al., 2023; Zandavi, Chung & Anaissi, 2021). In practice, these same engines can be pointed at MOOC-adjacent tasks (e.g., capstone presentations, live office hours, or assessment windows across regions), providing transparent trade-offs when equity (time-zone fairness) must be balanced against utilization and staffing cost.

Hybrid pipelines that combine evolutionary search with predictive models trained on platform data are also employed in Intelligent Tutoring Systems (ITS) integrated into large online courses. In institutional courses, interpretable genetic programming models have been trained on participation logs to forecast final performance and to surface human-readable rules that instructors can act on (e.g., low reciprocity in discussion forums + late quiz start → high risk), closing a loop between model insight and targeted support (Xing et al., 2015). Simultaneously, deep models on VLE big data have been implemented for early alerts in large cohorts; these models are used extensively to both select compact high-signal feature subsets and also tune the alert threshold, furthering local objectives such as minimizing advisor burden while maximizing recall (Waheed et al., 2020; Tan et al., 2017). These deployed systems demonstrate that evolutionary search may be a better complement to data-driven prediction, making decisions about how (and how aggressively) to intervene at scale.

A recurrent theme in real-world platforms is the need to juggle competing goals – completion, satisfaction, content quality, cost, and fairness – without collapsing them into a single metric. Production teams therefore adopt multi-objective evolutionary algorithms (MOEAs) to expose Pareto-efficient policy sets for course leads to review (Deb et al., 2002; Zhou et al., 2011). For instance, in adaptive modules deployed across large cohorts, MOEAs have been used to co-optimize the frequency of remedial loops, the difficulty step-size between items, and latency constraints from the platform, enabling course staff to select configurations that match pedagogical values and operational limits (Nogueira et al., 2024; Chen, 2009). Such deployments underscore that "optimization" in MOOCs and learning at scale is not merely algorithmic: it is sociotechnical in the sense that it requires auditable trade-offs for educator oversight.

Finally, at-scale retention efforts increasingly integrate evolutionary methods into early-warning and intervention services that run continuously during a course. Case studies in distance and open university settings report that GA- or EC-enhanced pipelines – combining feature selection, risk modeling, and playbook assignment – identify at-risk learners early enough to trigger timely outreach, with area-under-the-curve scores and operational impact reported on production cohorts rather than sandbox datasets (Zakaria, Anas & Oucamah, 2022; Queiroga et al., 2020; Hao, Gan & Zhu, 2022). Zakaria et al. (2022), for example, propose a machine learning approach that combines Genetic Algorithms and deep learning to early predict student dropout risk in MOOCs and enable personalized interventions, demonstrating improved performance over baseline methods across multiple datasets. By tying candidate interventions (extra practice, alternative media, peer support, or staff escalation) to explicit optimization objectives and platform constraints, these systems have moved from promising research into sustained service, improving equity of support while remaining transparent about the trade-offs selected by educators and program leads.

## 12.2 Personalized learning pathways

Personalized learning pathways are pivotal in enhancing student engagement and optimizing educational outcomes by tailoring content and instructional strategies to individual needs. This section delves into the dynamic implementation of adaptive learning systems within online environments, focusing on how advanced algorithms, such as Genetic Algorithms, facilitate the customization of learning experiences by creating work groups and personalized assessments, selecting learning content based on students' needs or features, and recognizing and proposing recommendations to students at risk. By integrating real-time performance metrics and evolving datasets, these methodologies ensure that educational approaches remain relevant and effective, ultimately contributing to equitable education across diverse learner populations. Following the discussions on personalized pathways, the paragraph will explore the critical applications and implications of these systems within broader educational contexts.

### 12.2.1 Implementation in online learning environments (case studies)

Below are field-tested deployments where GA/EC methods moved from theory to practice, embedded in live courses, LMS plug-ins, or institutional analytics pipelines, and were evaluated with real learners and operational constraints.

#### GA-enabled grouping in undergraduate courses

*Li, Ouyang, and Chen (2022)*. In Chinese undergraduate university classes delivered in an online learning environment, Li, Ouyang, and Chen (2022) propose a GA-based grouping method to address two main challenges: uneven group sizes and unknown students' features at the beginning of a course (Cold Start). Authors combine a Feature Categorization Model (FCM) to distinguish between static and dynamic (not known at the beginning) student characteristics, and a GA-enabled Insert Virtual Members strategy (IVMGA) to address uneven group sizes by adding virtual members to complete the group size. An N x M matrix is generated, where the rows correspond to the list of students and the columns correspond to the features analyzed. Candidate groupings are evaluated using a fitness function that aggregate multiple learner characteristics (demographics, communicative and collaborative skills, leadership ability, and participation level). The fitness formulation aims to promote heterogeneous composition within groups while maintaining balanced capability levels across groups. In a quasi-experimental design in online collaborative writing tasks, GA-generated groups are compared with randomly assigned groups and student-formed groups. Results show that GA-based grouping achieve significantly higher or more stable collaborative performance than random grouping, while performing comparably to student-formed groups, regardless of group size. Student perception data further indicate higher satisfaction with the level of conflict resolution in GA-generated groups.

*Moreno, Ovalle, and Vicari (2012)*. The study presents a genetic-algorithm-based method for forming collaborative learning groups. Each chromosome encodes a complete assignment of students to groups, and the fitness function is designed to simultaneously promote intra-group heterogeneity (to leverage complementary skills among peers) and inter-group homogeneity (to ensure comparable task difficulty across groups). Students are represented as numerical attribute vectors derived from diagnostic questionnaires measuring subject knowledge, communicative skills, and leadership abilities, all normalized to a common scale. The method was tested in real classroom deployments with 135 first-year university students at the National University of Colombia and compared against random (7) and self-organized (11) group formation strategies in a set of problem-based exercises. Results show that GA-formed groups (9) achieve significantly higher collaborative task grades than both baselines, while individual post-activity exam performances show no statistically significant differences. These findings indicate that GA-based grouping can effectively improve collaborative outcomes while remaining computationally feasible for realistic class sizes.

### Platforms for multi-criteria group formation

*Ugarte and colleagues (2022)*. The article presents TalSor, a platform designed to automate the formation of effective student teams. TalSor uses Genetic Algorithms to create well-balanced and heterogeneous teams based on gender balance, similar capacity and number of participants, and students' behavioural tendencies in team environments, as measured by the Belbin Team Role Self-Perception Inventory, which was preferred over personality tests. The platform allows instructors to specify the number or size of teams and gender balance, particularly relevant in engineering education. TalSor was evaluated in a real Chemical Engineering course at the University of the Basque Country using project-based learning. Results show that the generated teams are well balanced and are perceived by instructors and students as more effective than self-formed teams.

### LMS plug-in for quiz generation

*Fakhrusy and Widyani (2017)*. The authors present quiz generation as an optimization problem with soft constraints. In the paper, they discuss the development of Moodle plugins that enable quiz generation using Genetic Algorithms, where each individual is a candidate quiz, and each question represents a gene. The plugin helps select items from the question bank to approximate soft constraints such as total score, difficulty, time, and content coverage (number of questions per chapter). The implementation starts with the main Moodle Quiz plugin and several supporting question-type plugins; it is integrated into Moodle 3.1 and implemented in PHP. The fitness function measures how closely the generated quiz matches the specified constraints.

### Evolutionary personalization of content delivery at platform scale

*Christudas, Kirubakaran, and Thangaiah (2018).* The paper presents an evolutionary method for adapting content delivery in asynchronous e-Learning environments through the use of Genetic Algorithms. Personalization is framed as an optimization task in which learning objects are selected according to their compatibility with individual learner characteristics. In particular, the approach accounts for learning style, current knowledge level, and degree of interactivity, updating these parameters on the basis of (previous and actual) learners' feedback and interaction collected during the course. To improve efficiency, the authors introduce a modified genetic algorithm, Compatible Genetic Algorithm (CGA), which constrains the search space by enforcing compatibility between learning objects and learner styles. Learner profiles are modeled using the Felder-Silverman Learning Style Model. An experimental evaluation conducted with 240 participants in an online programming course, in which students autonomously chose 12 learning materials and received the other 12 via the recommendation system, shows that the proposed approach leads to higher learning outcomes and greater learner satisfaction, while also achieving better computational performance than a standard Genetic Algorithm and Particle Swarm Optimization.

### Adaptive objectives and paths via GA inside dynamic website

*Hssina and Erritali (2019)*. In this work, a genetic-algorithm-based approach is used to generate personalized learning paths aligned with educational objectives defined by the teacher. The adaptation problem is formulated as an optimization task. The learner's profile, expressed in terms of acquired and non-acquired concepts (binary representation), is updated through the initial assessment results. The Genetic Algorithm searches for an optimal sequence of learning modules that minimizes the distance between the learner's current profile and the target pedagogical objectives fixed by teacher. Fitness is defined based on the degree of alignment between acquired concepts and required objectives. The approach was tested with 29 learners.

### Program-level early risk surfacing from institutional data

*Sahlaoui and colleagues (2024)*. The article examines how advanced machine learning techniques can improve the prediction of student academic performance in higher education contexts characterized by limited and imbalanced data in Nigerian classrooms. The selection of features, including study time, parents' education levels, school support or absences, health status, and participation in extra-paid classes, plays a central role in the study. Genetic Algorithms, compared to other methods, are applied to identify the most relevant predictors while reducing dimensionality. To demonstrate practical applicability, the authors integrate best-performing models into an interactive web-based application built with Streamlit and Heroku. The application empowers institutions to access real-time results, personalize timely student support interventions, and enable policymakers to design data-driven decisions.

*Queiroga et al. (2020)*. In a Brazilian distance technical high school course lasting 103 weeks, the authors integrated students' interaction data from Moodle logs into a predictive pipeline aimed at identifying at-risk learners, generating prediction models on a bi-weekly basis during the first year of the course. Several machine learning classifiers are trained and evaluated, with their hyperparameters optimized using an elitist Genetic Algorithm in which the best solutions are directly carried over to the next generation. The study reports results comparing other techniques for hyperparameter selection and emphasizes early predictive performance rather than post-hoc analysis. It shows how Genetic Algorithms in hyperparameter tuning can achieve robust early discrimination between dropout and successful students, supporting the feasibility of deploying predictive analytics in real educational settings.

### MOOC-scale prediction with personalized recommendations

*Hao, Gan, and Zhu (2022)*. Tested on large-scale MOOC data from the OULAD (Open University Learning Analytics Dataset), the authors construct a Students' Performance Prediction Bayesian Network (SPBN) to predict students' final course outcomes by modeling uncertain and non-linear dependencies among demographic and engagement-related features. Based on these prediction results, a Genetic Algorithm (GA) is applied to generate personalized performance improvement suggestions for students at risk of failure. The Genetic Algorithm computes the minimum required engagement levels, specifically the number of clicks on learning materials and the average intermediate test scores, needed to reach the target performance of passing the course, and combines them with students' personal features.

### Implementation learnings across cases

The case studies show how GAs are used in educational settings as a standalone tool to solve optimization tasks as grouping, in combination with other techniques in pipelines to tune hyperparameters or generate solutions and recommendations, and in comparison with other methods to verify the more efficient procedure. Across deployments, several practices recur:

- expose and version constraints/fitness so instructors can see, edit, and justify optimization criteria;
- log decisions to enable audit and post-hoc analysis;
- evaluate on educational outcomes, students' skills, assessments, interaction, academic success, and features of learning content, not just loss curves.

## 12.3 Curriculum optimization tools

Effective curriculum optimization is essential for enhancing educational outcomes and ensuring that learning experiences meet the diverse needs of students.

This section presents tools and methods that use advanced evolutionary approaches to facilitate curriculum design and timetabling in higher education. More broadly within the discourse of data-driven educational research, we explain how these optimization tools can yield more efficient and inclusive learning paths that remain auditable and aligned with pedagogical intent.

For the sake of brevity, we do not report the applications listed in the previous paragraph, aimed at creating student grouping and content recommendation systems, which equally influence curriculum design.

### 12.3.1 Examples from education institutions

The adoption of the GAs in curriculum optimization in higher education helps to work out tangible solutions that improve personalized learning plan design, timetable assembly from multiple courses, and program-level guidance. In what follows below, we draw attention to research and real-world implementations and deployments that demonstrate not only the methodological possibilities inherent in GA/EC but also conditions for sustained institutional uptake (human-in-the-loop controls; auditability; fit with existing workflows).

#### Ontology-guided personalized sequencing in e-Learning system

A work by Chen (2009) on concept maps and ontology-driven sequencing, combined with GA search, shows how to personalize learning paths in elementary school while preserving prerequisite coherence. Chen (2009) modeled hierarchical relations among domain concepts on fractions and then used the structure to plan individualized curriculum sequences based on incorrect test responses from a pretest. In web-based learning environments, this can reduce cognitive overload and the “disorientation” that students often report when curriculum elements are poorly ordered. Within the GA search, structured constraints (difficulty of course material, concept relations of prior and posterior knowledge) are provided through the concept map to improve the logical coherence between course sequences.

Applying solutions like this in real cases in higher education, a key practical feature is explainability: the map encodes prerequisite logic, instructors can check why a given module is recommended, and modify constraints without rewriting the optimizer. This balance between algorithmic efficiency (the GA’s search over many feasible sequences) and human editability (instructor control over ontology/constraints) can be essential to classroom adoption and iteration.

#### Academic career-planning and pathway optimization prototypes

At the program design layer, GA-based expert systems can propose individualized module sequences that adhere to institutional constraints (prerequisites, capacity, term rules) and simultaneously target student-specific objectives. A GA prototype is illustrated by El Yessefi et al. (2024) that encodes modules as genes and optimizes module sequences, while ensuring logical continuity among them. The fitness function, in fact, evaluates the consistency between the competencies acquired at the end of each module and the prerequisite requirements of subsequent modules across semesters (chromosomes). The model was tested on real data from a Moroccan higher-education institution and validated using a satisfaction questionnaire administered to students and assessment records from 2017 onward. Allowing auditability (explanations of why a module is placed where it is), the approach exemplifies GA's potential as a decision-support tool for academic planning, which keeps humans in the loop and ensures that program-level changes remain pedagogically justified and institutionally feasible.

### Institution-scale timetabling under real constraints

Timetabling is a canonical curriculum-operations problem where institutions routinely face hard capacity, precedence, fairness, and availability constraints. A widely cited case by Daskalaki, Birbas, and Housos (2004) was validated on a real Engineering department. It formulated a full university timetabling instance as an integer program demonstrating conflict reduction and feasibility gains under hard and soft constraints (rooms, lecturer availability, teacher/student cohort overlaps, completeness of courses, and classroom changeovers). Although at the solver core there is integer programming (not a GA), this deployment remains instructive for the success hinged on explicit modeling of institutional rules and precise constraint modeling in solving complex educational scheduling problems.

Recently, a survey in the European Journal of Operational Research by Ceschia, Di Gaspero, and Schaerf (2023) identified six widely adopted formulations covering high-school, university course, and examination timetabling, discussing their structure, constraints, and practical relevance. They report the role and changes of both exact and (meta)heuristic techniques compared to exact methods. Across the six timetabling formulations, exact methods are mainly used to provide optimal solutions or bounds for small and simplified instances, while they struggle to scale to realistic settings. In contrast, metaheuristics – including genetic and evolutionary approaches – offer the flexibility and scalability needed to handle large, highly constrained timetabling problems and are therefore the dominant choice in practice.

Moving to GA, León and Castillo Domínguez (2021) presented an application of the use of Genetic Algorithms to improve the allocation of class schedules in a Peruvian technological higher education institution. The experimental results of their research demonstrate significant reductions in scheduling time and costs, as well as enhanced classroom usage. The study concludes that Genetic Algorithms can significantly improve administrative efficiency in academic timetabling.

For a domain-spanning synthesis focusing on exam timetabling methods and deployed systems, see Qu et al. (2009). One instance of an application in this sector is by Ngo et al. (2021). The authors present a GA-based approach to create viable exam schedules for large-scale instances with more than 2,500 students. The multi-objective model balances the number of students across exams, minimizes students' waiting time between the last exam and the next one, and reduces the number of exam rooms used, while satisfying strict hard constraints. Experiments conducted on real university data demonstrate stable convergence and good solution quality across multiple runs.

## Preference-aware student timetabling to improve choice quality

In recent years, the student course timetabling problem has changed from a feasibility-focused approach to a much more enriched concept that addresses new objectives such as students' preference satisfaction, fairness, and experience quality perceived. Mahlous and Mahlous (2023) proposed a Genetic Algorithm that explicitly encodes student preferences, including day and period preferences (soft constraints), while simultaneously ensuring compliance with institutional regulations and necessities (hard constraints: clashes, missing or incorrect allocations). The empirical evaluation using real-world datasets from King's College London showed that the proposed approach had very high feasibility and satisfaction with declared preferences (90% of students), thus significantly diminishing manual adjustments while improving overall satisfaction. At the institutional level, Dunke and Nickel (2023) took several steps further in simultaneously generating course timetables and individual student schedules using a multi-level matheuristic that not only considered conventional planning criteria but also incorporated student preferences.

Combined with the artificial neural network metamodel, NSGA-II–based multi-objective selection is applied in order to ensure a population of diverse non-dominated timetable solutions. This approach, between joint scheduling of lecture/tutorials and individual student preference, optimizes on multiple criteria to balance institutional feasibility with particular needs in complex university timetabling scenarios. Muklason et al. (2017) noted that exam timetabling quality should not only be based on technical feasibility but also on fairness over the student population, leading to multi-objective approach. Their extended formulations integrated preference satisfaction with equity considerations obtained from a student survey, showing that fairer timetables can be produced with limited impact on overall solution quality.

These studies encourage timetabling systems to account for configurable preference weighting and minimal-perturbation adjustments when schedules are changed, i.e., allowing schedulers explicit controls over soft-constraint weights and preserving stability when constraints change mid-term. Attention needs to be directed to modeling choices in constraints – how preferences are weighted and how solutions respond to changes – that is highly relevant for producing practically acceptable timetables (e.g., real-life instances in Müller, 2016; Müller & Rudová, 2016; Luo & Niu, 2024; survey in Ceschia et al., 2023). Collectively, these contributions underscore the need to treat "quality of experience" as a first-class objective in student timetabling, rather than as a secondary consideration.

## Multi-objective frameworks that match institutional trade-offs

Curriculum decisions are seldom made using a single criterion; institutions have to balance the speed of progression, coherence of prerequisites, equitable workload, and room capacity together with staffing considerations and fairness. The trade-offs are formalized in multi-objective EC frameworks. Many campus schedulers and prototype pathway tools summarized in Ceschia et al. (2023) are underpinned by NSGA-

II (Deb et al., 2002) and MOEA/D (Zhang & Li, 2007). The Pareto-front perspective is practically appealing to committees: decision-makers can explore non-dominated timetables or sequences, see how small limits on one metric (e.g., a few more slots in the evening for lecturers) can obtain large gains in another metric (e.g., a substantial reduction in student clashes), and select options which satisfy policy and equity commitments. The technical details are less important to stakeholders than the interface affordances: weight editors, fairness dashboards (e.g., 8 a.m. slot distribution across programs), and "what-if" scenarios for rapid consensus-building.

Studies by Cheong, Tan, and Veeravalli (2007), Lei, Shi, and Yan (2018), and Hafsa et al. (2025) frame examination and course timetabling as a multi-objective optimization problem, where multiple conflicting goals are addressed simultaneously. Multi-objective evolutionary algorithms are applied to balance trade-offs such as timetable length, student clashes, and room usage, producing sets of non-dominated solutions rather than a single optimum.

### Implementation takeaways across cases

Across these deployments, several patterns recur:

- *expose and version constraints/objectives.* Whether sequencing modules (Chen, 2009; El Yessefi et al., 2024) or scheduling lectures and exams (Ngo et al., 2021; Mahlous & Mahlous, 2023), successful systems enable instructors and schedulers to view and edit what the optimizer is targeting.
- *keep humans in the loop.* Adoption of tools and methods is narrowed by trust in systems that allow for rationales, alternative solutions, and an override process.
- *evaluate with educational outcomes.* In addition to solver metrics, assess clash reduction, progression coherence, participation balance, satisfaction, and persistence (Ceschia, Di Gaspero & Schaerf, 2023; El Yessefi et al., 2024; Mahlous & Mahlous, 2023).
- *plan for operation*s. Privacy, data freshness, and compute budgets matter. Staged pilots (e.g., one department, one term) are useful for tuning thresholds and weights ahead of scaling (Luo & Niu, 2024).
- *design for equity.* Preference-sensitive fitness terms (Mahlous & Mahlous, 2023) and fairness views (see Muklason et al., 2017) aim to ensure that advantages are not concentrated in already-advantaged cohorts.

On balance, these applications of Genetic Algorithms and Evolutionary Computation in higher education underscore their potential for improving curriculum organization and timetabling, thereby enhancing student achievement and performance. By addressing multifaceted constraints with transparent, auditable optimization, GA/EC methods support inclusive, efficient, and adaptable program design and delivery.

## 12.4 Resource allocation in distance learning

Resource allocation in distance learning is a multi-objective, high-stakes optimization problem. Institutions must continuously balance instructional capacity (instructor availability, tutor/TA coverage, grading and feedback bandwidth), slots for synchronous sessions (also across time zones), virtual classroom concurrency, assessment proctoring windows, content delivery infrastructure, and targeted student support. These demands spike and shift by cohort, week in term, and even hour of day while equity, privacy, and cost constraints tighten.

The Open University (OU) has operated one of the most well-known, production-grade analytics ecosystems supporting distance cohorts. The Open University Learning Analytics Dataset (OULAD) captures anonymized VLE traces made available to everyone (Kuzilek, Hlosta & Zdrahal, 2017; Hlosta et al., 2018). This dataset is linked to OU Analyse, an early-warning service based on machine learning methods and used by tutors and student support teams to classify students and predict those at risk (see analyse.kmi.open.ac.uk). Critically, this approach pairs predictive models with operational allocation. Alerts are converted into prioritized advising queues for tutors who can intervene for at-risk students. The workflow allows staff to verify alerts and choose the right support for students, creating a human-in-the-loop feedback loop that enables tuning of precision over time. The process focuses on prediction and tutoring and does not adopt GAs, but represents a guideline to apply live interventions that improve students' retention and allocate available human resources.

In settings like this, Genetic Algorithms (GAs) and Evolutionary Computation (EC) may offer outer loops that explore complex assignment spaces, balance conflicting objectives, and adapt in near-real time as signals arrive from LMS/VLE logs, streaming video analytics, and advising systems.

Practical patterns underlie successful deployments: (i) expose optimization criteria clarifying what fitness means operationally; (ii) match EC with robust learning analytics pipelines to keep data fresh, (iii) align evaluation with learner-centric outcomes such as persistence, tutor response times, equitable access, and (iv) design for auditability and human-in-the-loop overrides. The following case-led section synthesizes how researchers have used GAs/EC to allocate resources at scale and, sometimes, day-to-day academic operations. Some of the cases described earlier, in the previous paragraphs, will be revisited and analyzed from the perspective of resource allocation.

### 12.4.1 Case study: optimization in universities and research

#### Case A - Allocation of human and physical resources in real and online educational spaces

The study by He (2022) identifies a set of indicators for measuring resource utilization in universities and proposes a Genetic Algorithm to improve the efficiency of this process. The resources have been aggregated into three categories: human resources, including the number of full-time and part-time teachers, administrative staff, and non-academic external faculty; physical resources, measured by indicators such as the value of teaching tools and equipment and the room area; and special funding investments, representing financial resources. The fitness function compares these input resources with output indicators related to talent cultivation (number of trained students, number of enterprises founded by students, and number of prizes acquired in academic or provincial contests), scientific research (number of publications and research projects), and social service (production value of scientific and technological achievements). Each indicator is assigned a weight validated through expert judgment reflecting its influence on educational outcomes. Experimental results show that the efficiency of utilization and allocation of educational resources in colleges and universities involved in the study increased, respectively, by 21.33% and 18.92% on average. Similar to this is the work of Qiu (2022), who combines Genetic Algorithms and Particle Swarm Optimization (PSO) in the resource allocation process. There are examples in the literature where Genetic Algorithms are used to identify optimal sites for the construction of education centers based on literacy rates and property costs (Agrawal, Agarwal & Bansal, 2020), and to design comfortable indoor environments that consider both energy and cost (Xu et al., 2021).

Finally, dynamic resource allocation is considered in MOEA as the mechanism that allows optimizing multiple objectives simultaneously and assigning more computational resources to the more promising solutions, as explained in Zhang et al. (2024), who present a framework combining educational data mining, learning analytics, and evolutionary computation techniques for higher education management systems.

*What was allocated?* Human, physical, financial, computational and energetic resources in the management of educational institutions.

*Why does it matter?* The development of educational institutions and teaching practices, whether online or face-to-face, is closely linked to how resources are utilized, financed, and shared.

### Case B - Preference-aware timetabling to balance students' demand and participation (impact on large online programs)

Preference- and constraint-aware timetabling – long studied in on-campus settings – has been adapted to distance timetables, instructor availability, and capacity of real-time virtual rooms. The research by Mahlous and Mahlous (2023), already illustrated in section 12.3, describes a GA that explicitly incorporates the student preference and institutional constraints (e.g., avoiding clashes with other courses or observing protected time windows for working students), leading to improved student satisfaction and downstream engagement. While the research context is not purely online, the mechanism can be directly applied to distance education, where equal shares of synchronous and asynchronous sessions and proctored assessment windows are essential. Importantly, building on this study, we can conclude that timetabling optimization starting from students' choices is possible within actual workflows and results in better outcomes than heuristic or manual approaches.

*What was allocated?* Session slots (or assessment windows), balancing expressed student preferences with instructor and platform constraints.

*Why does it matter?* In distance programs, time-zone equity and life-schedule compatibility are a core retention lever; GA search lets institutions treat fairness as a first-class objective.

## Case C - Video learning analytics to steer supplemental instruction resources (online courses at scale)

At scale, video is a dominant learning surface in distance programs. Hasan et al. (2020) analyze video learning interactions (play, pause, seek, and replay) in a mobile learning application integrated with an LMS to predict learners' academic performance. In this work, GAs are employed as a tool for feature selection to train classification models using students' information, time spent on Moodle, and video interactions. Machine learning methods are assessed, with Random Forest calculating the best predictive precision.

The takeaway here in terms of empirical practice: video trace analytics can translate into staffing and content-production queues that meet students where the actual friction is.

Video interaction data provide valuable signals for identifying students at risk of poor performance. Institutions could use these signals to allocate supplemental help: teacher assistants (TAs) can generate micro-questionnaires after worrisome videos or add remedial content for students experiencing similar difficulties. The takeaway here in terms of empirical practice is that video trace analytics can be translated into actions for staff and content production that meet students where and when the trouble actually is.

*What was allocated?* TA office hours, micro-quiz authoring bandwidth, and instructor revision time, targeted to the "pain points" surfaced by video traces.

*Why does it matter?* In distance learning, there is no hallway conversation to reveal confusion; analytics-guided allocation constitutes a data-driven compass.

## Case D - Program-level pathway planning to structure teaching offering

At the program level, institutions need to consider the whole made by prerequisite and learning outcomes, semesterly capacity, and TAs availability to support individual student progress. We have already described the study by El Yessefi et al. (2024), which reports a GA-based expert system for sequencing academic pathways, where modules are treated as "genes" and constraints, such as prerequisites to satisfy and competence to acquire, are included. As a decision-support prototype with actual programs in use, the system suggests content sequencing that reduces inconsistent module ordering and preserves academic progression. The design encourages fitness transparency (why now for this module?), which is key to making adoption recommendations. Adapting (or customizing) the structure of a program delivered online or in person to students' features/goals and institutions' organizational structure can be considered an allocation action, in which the training events, course delivery, and support from teachers and tutors represent resources built up throughout a student's training path.

*What was allocated?* Module slots, aligning student pathways with institutional capacity and course prerequisites.

*Why does it matter?* Student cohorts are heterogeneous and mobile; pathway optimization mitigates choke points (as "killer exams") without forcing one-size-fits-all progressions.

### Case E - AI in online higher education: integrating predictive models with service capacity

A systematic review of empirical studies on artificial intelligence use in online higher education (Ouyang, Zheng & Jiao, 2022) synthesizes how AI is applied towards predicting student performance or dropout risk, as well as recommending learning resources, and relies heavily on process data such as learning management system activity, assessment submission, and login behavior. These predictive and recommendation models are primarily implemented to assist with the early identification of at-risk students and to provide timely feedback that enables corrective action for learners and instructors. Advanced optimization approaches, such as Genetic Algorithms, appear only rarely in the reviewed literature.

Building on this work, we can underline the strict relation between predictive outputs and concrete allocation decisions, such as prioritizing student outreach or targeting limited support resources, especially in large-scale online programs where human support capacity is finite. From this perspective, the operational challenge can be viewed as a resource allocation problem, where institutions must balance competing objectives such as maximizing expected student success, minimizing staff overload, and ensuring equitable access to support.

*What was allocated?* Human services (advising, tutoring, help-desk time) and adaptive content slots (recommendation "real estate") to the students most likely to benefit.

*Why does it matter?* In distance settings, service capacity, together with algorithmic accuracy, determines whether early warnings convert into outcomes.

### Case F - Group formation and peer-support as a resource multiplier in online courses

Distance programs also allocate peer support as a complement to instructor presence. In section 12.2, we have analyzed implementations (see Li, Ouyang & Chen, 2022; Moreno, Ovalle & Vicari, 2012) that show how GAs can improve this kind of resource allocation related to group creation. The approach considers multiple characteristics of learners, including demographics and skills, to foster group compositions that are sometimes homogeneous and sometimes heterogeneous, while, in the latter case, ensuring balanced intra-group diversity and inter-group comparability.

*What was allocated?* Peer-support "capacity" (who learns with whom) that would reduce instructor load, maximize collective efficacy, especially relevant to fully online projects.

*Why does it matter?* In distance formats, well-formed groups replace lab buddies and can affect motivation and throughput.

## Case G - Multi-objective frameworks that scale with modality growth

In conclusion, generally speaking, when cohorts and offerings grow, and online solutions and environments occur, allocation quickly becomes multi-objective: minimize conflicts between cohorts/classes/day/time; respect work-rules of faculty, tutors, and staff; maximize fairness with regards to needs for all participants; ensure instructor continuity; manage platform concurrency, and so on. The NSGA-II and MOEA/D families are the workhorses many teams lean on when a single scalar objective is too crude (Deb et al., 2002; Zhang & Li, 2007).

*What was allocated?* Everything at once-teacher assignments, session slots across time zones, platform concurrency, and equity constraints-surfaced as Pareto-optimal alternatives for human selection.

*Why does it matter?* Pareto sets expose trade-offs, enabling accountable decisions.

## Cross-cutting operational lessons

Across deployments, several practices recur.

- *Expose and version fitness functions.* Successful solutions let see and edit what "good" means in fitness function (conflict penalties, fairness weights), treating optimization criteria as policy, not a black box (He, 2022; Hasan et al., 2020).
- *Design for auditability.* Record why a student was flagged, why a proctoring slot was assigned, or why a group was formed; this is essential for trust and for post-hoc bias checks.
- *Tie models to capacity.* Early-warning in the absence of advisor capacity merely reorders risk; implementations that batch alerts and cap caseloads show better durability (Kuzilek, Hlosta & Zdrahal, 2017).
- *Favor lightweight pilots.* Invest in one program, one course family, or one tutoring queue; work with learner-centered metrics (persistence, response time, fairness indices) instead of algorithm-centric scores.
- *Use hybrid loops.* Integrate predictive models (video analytics, grades predictions) when relevant with an EC outer loop that optimizes scarce human and infrastructure resources based on explicit constraints (Hasan et al., 2020; Qiu, 2022).

# Chapter 13

# Educational Data Mining and Evolutionary Algorithms

The data boom has impacted many fields, including education, requiring institutions and researchers to attempt to improve student learning paths and optimize learning environments through a data-driven approach. Two fields of investigation have emerged since the early years of the new millennium: Educational Data Mining (EDM) that uses advanced computational techniques to extract useful information from educational data (Baker & Inventado, 2014), and Learning Analytics, closely related to the previous one but differing in the degree of human involvement, investigation methods, and the reductionist or holistic approach (Siemens & Baker, 2012).

This section examines the powerful intersection of Educational Data Mining (EDM) and Evolutionary Algorithms, highlighting their applications in classification, clustering, and predictive modeling. By examining the capabilities of these methodologies, readers will gain an understanding of how they can effectively drive personalized learning pathways, improve resource allocation, and inform decision-making within educational contexts. This exploration delves into the possibilities that data science offers as part of a larger narrative on bringing sophisticated computational methods to education.

## 13.1 Overview of EDM techniques and trends

In their systematic review on the use of soft computing in Educational Data Mining (EDM) and Learning Analytics (LA), Charitopoulos, Rangoussi and Koulouriotis (2020) found that approximately 70% of the studies published between 2010 and 2018 in the EDM field used soft computing methods. In contrast, in Learning Analytics only about 22% of studies adopted soft computing approaches in the same period. In Educational Data Mining research, the main topics investigated using soft computing techniques include the prediction of learning outcomes and dropout rates, university acceptance and enrollment, and the optimization of learning material sequencing. The predominant application context is e-Learning environments, including Learning Management Systems (LMSs) and Virtual Learning Environments (VLEs). However, traditional educational settings are also represented in the literature, including non-computerized contexts and studies based on data extracted from departmental or faculty registries, student demographic information, and historical grade records from previous academic years. Among soft computing approaches, Decision Trees, Bayesian or probabilistic reasoning methods, Artificial Neural Networks (ANNs), and Support Vector Machines (SVMs) are the most widely used. Natural Language Processing and Genetic Algorithms (or Evolutionary Computing) are less frequently used overall, although they are preferred for specific types of tasks. In particular, Genetic Algorithms and Fuzzy Logic are mainly applied to classification and clustering problems, where they represent comparatively less popular methodological choices. The review underlines that soft computing methods are often employed either within or alongside more traditional analytical approaches. These findings leave wide opportunities for experimentation with evolutionary computation in the fields of EDM and LA.

Educational Data Mining (EDM) employs advanced computational techniques to derive actionable insights from educational data. Evolutionary algorithms, including Genetic Algorithms (GAs), can be utilized to address complex challenges such as classification, regression, clustering, and pattern mining, contributing significantly to the advancement of education (Baker & Inventado, 2014; Romero & Ventura, 2010; Charitopoulos, Rangoussi & Koulouriotis, 2020). A typical EDM pipeline with explicit EC contributions is shown in Figure 13.1.

Around 2008-2009, Alcalá-Fdez and colleagues presented a Java open source software named KEEL to analyze the behavior of evolutionary approaches and soft computing in data mining problems, such as regression, classification, and unsupervised learning. Although the last update was in 2015 (version 3), the design of this software, specifically created to merge data mining with evolutionary methods, remains valuable and informative. KEEL, which stands for Knowledge Extraction based on Evolutionary Learning, is especially influential because it provides a complete, research-grade environment to design, execute, and fairly compare EDM experiments that use

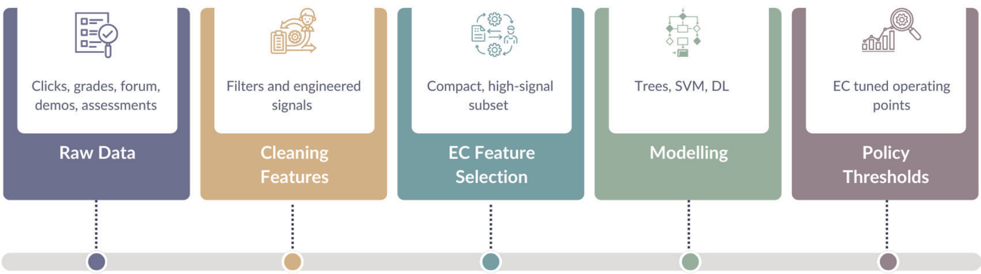


**Figure 13.1** - Educational Data Mining pipeline

EDM pipeline with explicit EC touchpoints: feature selection and policy/threshold optimization surrounding modeling.

evolutionary computation. Concretely, KEEL offers (a) a comprehensive library of data preprocessing operators (imputation, normalization, discretization, resampling for class imbalance, instance/feature selection), (b) many evolutionary learners (e.g., evolutionary fuzzy rule-based systems, genetic programming, ant-colony rule learners such as Ant-Miner), (c) XML experiment specifications that guarantee reproducibility (pipelines, parameter grids, random seeds), and (d) rigorous statistical evaluation (parametric and non-parametric procedures like Friedman/Nemenyi) to benchmark algorithms across multiple datasets and metrics (Alcalá-Fdez et al., 2009, 2011).

Two properties make KEEL particularly suitable for education. First, it emphasizes interpretable models (rule sets, fuzzy rules), so advisors and instructors can audit why a learner was flagged or a path was recommended; second, it supports structured experimentation that is critical when institutional stakeholders need evidence that a chosen pipeline is robust, fair, and reproducible before deployment.

By simplifying these processes, KEEL offers evolutionary learners, e.g., evolutionary fuzzy rule-based systems. Typical early-warning workflows combine imbalance-aware preprocessing (e.g., resampling) available in KEEL with evolutionary feature selection (Xue et al., 2016; Alcalá-Fdez et al., 2009, 2011) and a rule-based learner. The process is then validated through repeated cross-validation and statistical tests. This setup helps teams balance predictive performance with explainability, which is indispensable where model outputs inform advising or access to resources (Alcalá-Fdez et al., 2009). KEEL's curated repositories, experiment descriptors, and standardized evaluation allow for greater replicability and extension across labs and beyond institutions (Alcalá-Fdez et al., 2011).

Some tasks, trends, and key points can be described in EDM applications using evolutionary strategies.

*Handling class imbalance and missingness*. Common targets like dropout or failure in EDM are sparse events, which can make naive accuracy potentially misleading. Teams frequently bundle standard resampling pipelining (i.e., synthetic minority oversampling) or cost-sensitive losses with evolutionary wrapper-based feature selection, and assess models using AUC-PR (average precision) along with ROC-AUC to mitigate optimistic results under skew (Alcalá-Fdez et al., 2009; Rastrollo-Guerrero, Gómez-Pulido & Durán-Domínguez, 2020). The joint pipelines and evolutionary wrappers thus facilitate the identification of robust signal subsets that are minimally affected by gaps in the logs, especially when working with engagement traces (clickstream, VLE logs) for which systematic missingness is common, while preprocessing modules ensure regularized imputation across experiments (Alcalá-Fdez et al., 2011).

*Multi-objective optimization (MOO)*. Institutions have to balance conflicting goals: increase recall of at-risk students, reduce false alarms that burden advisors, maximize lead time for interventions, and maintain equity across subgroups. Pareto-based evolutionary algorithms such as NSGA-II and MOEA/D explicitly model these trade-offs, returning a frontier of viable policies rather than a brittle single optimum (Deb et al., 2002; Zhang & Li, 2007; Zhou et al., 2011). In practice, this lets stakeholders choose a point (a solution) in the front that best matches advising capacity or institutional priorities (e.g., a moderate-recall/low-burden policy during peak exam periods). Similar MOO concepts underlie curriculum sequencing that balances learning gains against cognitive load and time-on-task within high-level institutional constraints (Tan et al., 2012; Nogueira et al., 2024).

*Rule generation and interpretability*. Evolutionary, rule-based learners (e.g., Ant-Miner, Parpinelli, Lopes & Freitas, 2002) yield human-readable policies that can be acted upon by advisors. Evolutionary fuzzy systems likewise produce linguistic rules ("IF low forum activity AND declining quiz scores THEN high risk") that are auditable and can be mapped to interventions (study skills workshop, office-hours outreach). Such models are competitive, while also increasing transparency, a necessary condition for procedural fairness and effective human-in-the-loop decision-making.

*Hybrid AI + EC pipelines*. Evolutionary search works well as an outer loop for feature selection or hyperparameter/architecture search, with a gradient-based or tree-based learner as the inner loop. This hybridization often reaches state-of-the-art accuracy while keeping interpretability levers (e.g., feature sparsity, monotonicity constraints) in view (Bansal, Goyal & Choudhary, 2022; Alcalá-Fdez et al., 2009). Simultaneously, genetic programming has been applied to interpretable trajectory modelling of performance prediction and participation (Xing et al., 2015), which aligns with educational needs to explain and justify flags or recommendations.

*Temporal and streaming considerations*. Educational processes are sequences and, in online environments, features considered in the analysis or developmental processes are configured as event streams, for instance, logins and video interactions. Temporal cross-validation through rolling windows is used to prevent label leakage and to test how early a signal forecasts risk. In prequential (test-then-train) evaluation, each batch is scored by the models prior to learning from it; this simulates real-time deployments in VLEs/LMSs. Evolutionary wrappers can also optimize window sizes, decay rates, and aggregation functions for temporal features – parameters that are the key performance drivers in practice.

*Trend and cohort analysis*. Latent trends, for example, week-to-week engagement decay, become visible via longitudinal modeling (Queiroga et al., 2020). Evolutionary feature selection (Xue et al., 2016) helps isolate stable, actionable features across cohorts, improving robustness under curricular or platform changes. Performance gains are strongest when signals are interpretable, timeliness is explicitly optimized, and evaluation reflects operational constraints (advising capacity, intervention costs) (Albreiki, Zaki & Alashwal, 2021; Ifenthaler & Yau, 2020).

*Instrumentation and data standards*. Reliable EDM starts with high-quality event data that needs to be collected and stored. Two widely adopted, vendor-neutral standards are xAPI (Experience API) and Caliper Analytics. xAPI defines a flexible JSON statement model for learning events ("actor-verb-object"), enabling cross-platform traces (Advance Distributed Learning Initiative, 2016). 1EdTech Caliper (2020) defines well-typed event profiles (AssessmentEvent, MediaEvent etc.) and a telemetry architecture for higher education at scale. Setting ingestion to these standards makes feature engineering, model portability, and governance easier.

*Reproducible benchmarks and open datasets*. Public resources – as the Open University Learning Analytics Dataset (OULAD - Kuzilek, Hlosta & Zdrahal, 2017) – have accelerated progress by supporting replicable early-warning research and fair algorithmic comparisons before institutional rollout. So, it becomes possible to prototype in KEEL or other suites on OULAD or other open datasets, then transfer lessons to one's own institutional situations.

*Ethics, fairness, and governance*. Because EDM models affect people, their evaluation must go beyond accuracy. Practices have to include (a) subgroup audits (demographics, initial proficiency levels), (b) opportunity parity and fairness diagnostics (e.g., equalized odds, equal opportunity), and (c) calibration checks to ensure risk scores mean the same thing across groups (Hardt, Price & Srebro, 2016; Slade & Prinsloo, 2013). The DELICATE framework in learning analytics outlines how to embed privacy and ethics through eight practical principles (Determine, Explain, Legitimate, Involve, Consent, Anonymize, Tech, Evaluate), describing strategies for documenting and sharing purposes, legal bases, and stakeholder involvement (Drachsler & Greller, 2016). Reviews synthesize operational lessons learned about governance and responsible analytics in higher education (Papamitsiou & Economides, 2014; Ifenthaler & Yau, 2020; Bellini et al., 2019).

## 13.2 Applications of GAs and EC in data mining for education

Genetic Algorithms (GAs) and Evolutionary Computation (EC) have shown applicability for EDM issues which include classification, clustering, regression, and predictive modeling. These techniques enable educators and institutions to analyze extensive and complex datasets while optimizing outcomes.

In *classification* tasks, GAs play a crucial role by enhancing feature selection and parameter tuning. Queiroga et al. (2020) present a clear example of how a Genetic Algorithm can be used to tune classifiers (ML predictive models of dropout) and their hyperparameters based on data analyzed every two weeks using AUROC. GAs can improve classification rules and adapt to the dynamic, evolving features of educational data, thereby maintaining accurate results. This adaptability contrasts with traditional approaches, which often struggle to accommodate the varied and constantly changing variables in educational environments. In fact, GAs can outperform static models in scenarios where student behaviors and performance metrics change over time, highlighting their relevance for achieving real-time insights.

In particular, wrapper-style GA selectors (as implemented in research tooling such as KEEL) search the space of feature subsets while optimizing for metrics aligned to the educational objective (e.g., recall of at-risk students with an alert-fatigue constraint). This approach is especially valuable under class imbalance – common in the classification problem when dropout or failure events are rare – where AUC-PR (average precision) and recall are emphasized over raw accuracy (Alcalá-Fdez et al., 2009, 2011; Queiroga et al., 2020; Zhang et al., 2024). Besides, evolutionary feature selection and tuning can improve early-warning models' robustness when logs are noisy, signals drift across semesters, or data are missing non-randomly. In such pipelines, GAs also tune thresholds and decision costs – for example, prioritizing sensitivity during week-progress checks while capping the number of weekly alerts to match advising capacity.

Soft-computing reviews (Charitopoulos, Rangoussi & Koulouriotis, 2020) document EC-enhanced *clustering* applications (e.g., GA-initialized k-means, evolutionary fuzzy clustering) based on heterogeneous, partially observed engagement data, capable of returning interpretable groupings that can guide interventions, for instance, distinguishing among steady-engagers, late-surgers, and early-risk students. These clusterings surface latent trajectories that static algorithms can miss, informing timely nudges and resource allocation in large cohorts. A practical extension is cohort-aware clustering, wherein evolutionary objectives reward the stability of clusters across offerings to facilitate operational adoption and penalize subgroup disparities to guard equity. In Figure 13.2, we report an example of a 2D projection of clusters that identify three kinds of student behavior.

In *regression* applications, GAs excel at optimizing variable selection and reducing noise within datasets, contributing to bringing better performance prediction. See Figure 13.3 for a typical flowchart of GA applications in feature selection. Classic overviews (Beasley, Bull & Martin, 1993; Mitchell, 1995; Goldberg, 1989) describe how GAs search continuous parameter spaces and symbolic model forms. In EDM, this manifests as symbolic regression via genetic programming to model grade trajectories or participation-to-performance mappings with interpretable expressions (Xing et al., 2015). Compared with parametric regressions, evolutionary regression tolerates nonlinearity, interactions, and non-Gaussian noise, while GA wrappers suppress redundant telemetry features that otherwise destabilize forecasts (Alcalá-Fdez et al., 2009; Scrucca, 2013, 2017). Scalability remains a challenge: MOOC-scale datasets require incremental evaluation (prequential testing) and resource-aware search. Here, evolutionary pipelines benefit from early-stopping, island models, and parallelization to keep compute budgets tractable (Scrucca, 2017).

GAs have been applied in *rule generation*, where they can improve decision-making for personalized courseware (we'll discuss this in more detail in the next paragraph). Beyond evolving opaque parameters, EC can produce human-readable rules that educators and advisors understand, trust, and use to act. Ant-colony rule learners such as Ant-Miner (Parpinelli, Lopes & Freitas, 2002) and evolutionary fuzzy rule-based systems (available in KEEL) induce compact, interpretable policies that map to concrete actions (Alcalá-Fdez et al., 2009, 2011). An example to clarify: the rule

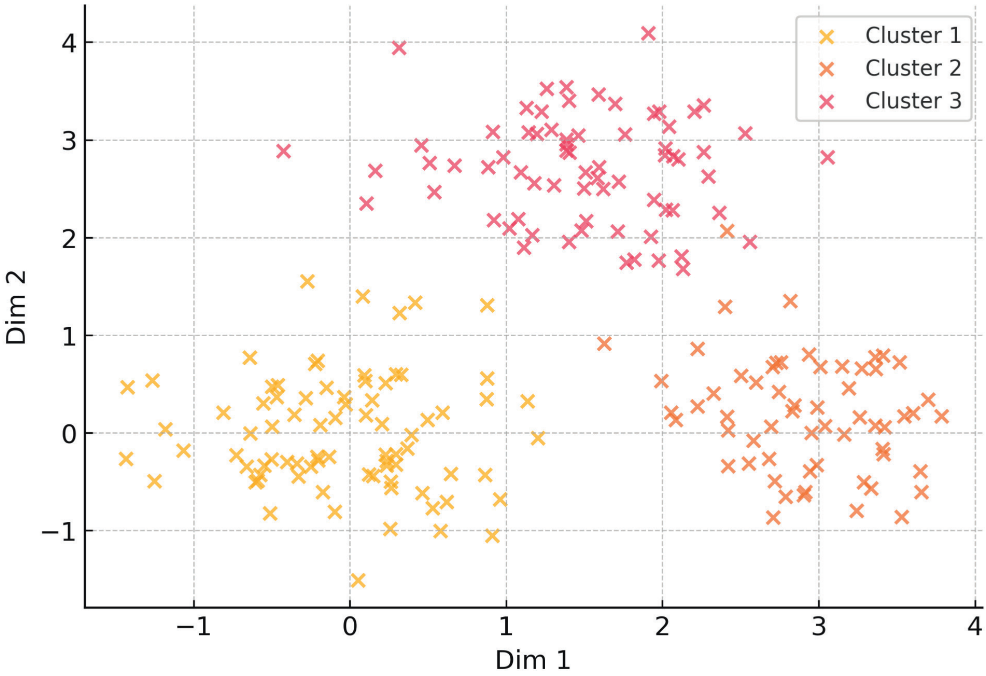


**Figure 13.2** - An example of Cluster Mapping

The 2D Projection of clustering illustrates behavioral clusters supporting cohort-level insights and targeted interventions.

"IF (declining quiz scores) AND (few forum posts) THEN (high risk)" implies that, given certain conditions regarding scores and participation levels, a student may be considered at risk and thus actions to improve retention are required. Complementary research in EDM by Sreenath and Jeyakumar (2016) demonstrates evolutionary rule discoveries (in details Genetic Algorithm, Genetic Programming and Differential Evolution) related to personalized learning object sequencing and courseware con-

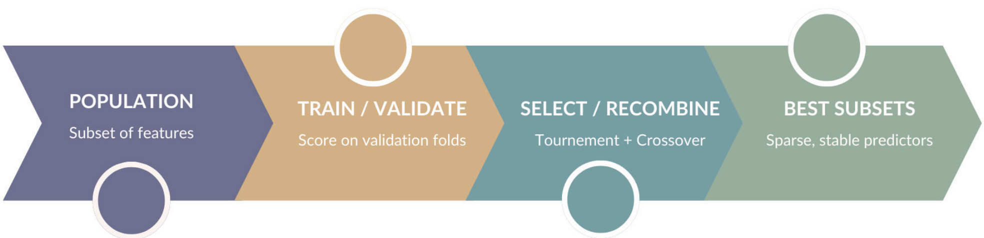


**Figure 13.3** - Genetic Algorithm wrapper for feature selection

GA wrapper for feature selection: population of subsets, cross-validated scoring, and recombination toward sparse, stable models.

struction, which are relevant to instructors' need to understand why particular modules were recommended. These evolutionary approaches make teachers' and institutions' choices more transparent and auditable, and help improve procedures for institutional review and ethics oversight. Research explores sparsity and monotonicity constraints to further improve interpretability without sacrificing accuracy (see Murphy, 2012).

The integration of evolutionary approaches with artificial intelligence (AI), such as neural networks, has led to advanced predictive models. A pattern (already discussed elsewhere in this book) is to use a GA as an outer optimizer for feature selection, architecture search, or hyperparameter tuning together with deep or gradient-boosted learners, merging the exploratory power of EC with the function-approximation strength of modern ML (Bansal, Goyal & Choudhary, 2022). Historically, GAs have tuned network topologies and weights (Harp, Samad & Guha, 1989; Stanley & Miikkulainen, 2002), and broader EC has informed policy search in reinforcement learning (Moriarty, Schultz & Grefenstette, 1999). In Learning Analytics at scale, hybrid GA+DL setups have improved prediction from complex VLE streams (Waheed et al., 2020), while leaving in place interpretability levers (e.g., sparse feature masks, post-hoc rule extraction) needed for high-stakes advising. Practically, such hybrids are commonly staged: prototype with evolutionary search on open datasets like OULAD, harden pipelines with reproducible experiment descriptors (KEEL-style), then move to institutional data with drift monitoring and fairness auditing.

The iterative mechanisms of GAs – fitness evaluation, selection, crossover, and mutation – ensure consistent refinement of models, enhancing their applicability in dynamically changing educational environments. Fitness functions in EDM typically encode educationally meaningful metrics (e.g., week-ahead recall of at-risk flags), sometimes combined with structural penalties (model size, number of features) to promote interpretability and stability. Selection pressure, controlled mutation rates, and diversity preservation help prevent over-fitted search to one cohort's attributes (Mitchell, 1995; Beasley, Bull & Martin, 1993; Deb et al., 2002).

Besides, within evolutionary methods, competing objectives in EDM require Multi-Objective Optimization (MOO) to balance minimizing dropout, maximizing student satisfaction, reducing advising workload, and ensuring subgroup equity. Pareto-front algorithms like NSGA-II and MOEA/D efficiently explore these trade-offs and return a portfolio of policies for human selection (Deb et al., 2002; Zhang & Li, 2007; Zhou et al., 2011). In education, MOO underpins (a) early-warning systems that jointly optimize recall, precision, and lead time (Hao, Gan & Zhun, 2022); (b) curriculum/adaptive pathing that balances mastery gains, cognitive load, and time-on-task (Tan et al., 2012; Nogueira et al., 2024); and (c) intervention targeting that trades off benefit to the learner against capacity constraints (Hafsa et al., 2025; Ngo et al., 2021). Setting these objectives avoids the optimization of single metrics, which can lead to brittle or inequitable deployments that are, moreover, poorly suited to educational realities. In practice, the added value of applying multi-objective solutions lies in the fact that the Pareto set can be filtered through operational constraints (e.g., advisor-to-student ratios) and ethical requirements that ensure subgroups perform similarly, before a policy proceeds to rollout for approval by those responsible for the process.

Two cross-cutting application themes need to become standard practice in real deployments:

- *temporal/streaming evaluation.* To avoid label leakage and measure true lead time, application teams are increasingly turning to prequential (test-then-train) evaluation and rolling-window validation since educational data is received over time and continuously updated. Evolutionary search often includes temporal hyperparameters (window size, decay) directly in the chromosome, allowing the optimizer to select representations generalizing across cohorts (Albreiki, Zaki & Alashwal, 2021).
- *instrumentation, fairness, and governance.* Reliable, cross-platform telemetry via xAPI and Caliper stabilizes features and eases portability across systems (Advanced Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; 1EdTech Consortium, 2020). In addition to accuracy, mature EDM now has to audit fairness and check subgroup calibration; if disparities are found, objectives or costs are adjusted (Hardt, Price & Srebro, 2016; Slade & Prinsloo, 2013). Evolutionary multi-objective framing intrinsically accommodates fairness metrics as explicit optimization targets, rather than after-the-fact add-ons.

Overall, at the end of this paragraph, a pragmatic roadmap can be described as follows: prototype with interpretable, evolutionary pipelines on open benchmarks; validate with temporal and fairness-aware evaluation; select Pareto-optimal policies aligned to advising capacity; and deploy with monitoring, drift detection, and governance that keeps people in the loop.

## 13.3 Rule generation and personalized courseware

Rule generation and personalized courseware construction are key developments in GA applications to the fields of Learning Analytics (LA) and Educational Data Mining (EDM), where GAs (and other evolutionary techniques), as shown by Sreenath and Jeyakumar (2016), can be used to derive decision-making rules from educational data and support the design of adaptive learning materials tailored to learners' characteristics and learning styles in a process that is also advantageous for institutions. GAs employ fitness functions to iteratively refine feature subsets and rule structures from educational datasets, ensuring that the generated decision-making rules align with learners' preferences, prior knowledge, and demonstrated abilities. Evolved rules, for example, can indicate the best chain of units to show next according to a mastery profile, or vary feedback granularity and timing depending on a student's recent correctness and latency metrics, thereby orchestrating both what is shown next and how support is delivered. Since this search is population-based and iterative, the learned policies can adapt to changes in cohorts, curricula, or platforms, increasing the utility for programs that utilize multiple modalities (on-campus/blended/fully online), some of which show considerable differences in engagement across terms. A practical challenge remains getting these systems in the hands of teachers, especially for those with limited technical skills; as a result, contemporary deployable solutions greatly prefer rule learners, which output compact (and therefore human-readable) rules and are paired with explanation panels and "what-if" simulators mapping their corresponding behaviors to concrete pedagogical actions. A study that applies GP to extract rules in a more accurate and understandable format than other techniques for building a prediction performance model is by Xing et al. (2015).

The integration of evolutionary methods – especially ant-colony and genetic rule learners – generates effective classification rules from educational datasets. In particular, Ant-Miner (a bio-inspired ant-colony algorithm) has proven adequate in identifying compact, actionable and interpretable rules in noisy, high-dimensional data, that is a common profile for clickstream and assessment traces (Parpinelli, Lopes & Freitas, 2002).

Within KEEL, together with Ant-Miner, researchers rely on evolutionary fuzzy and GA-based rule learners (Alcalá-Fdez et al., 2009, 2011). These learners can encode constraints that matter in education, for instance, penalizing overly complex rule antecedents to keep advising dashboards readable, or enforcing monotonicity (e.g., "lower attendance should not predict higher mastery" when all other things are equal). Comparative testing reported in KEEL's benchmark suites shows that evolutionary rule learners remain competitive in predictive performance while producing models that instructors can audit and edit, a prerequisite for responsible use in high-value applications like progression decisions and early-warning triage (Alcalá-Fdez et al., 2009, 2011). When working with large distributed systems (e.g., MOOCs), teams generally adopt island models and parallel evaluation to keep these learners tractable, and they observe temporal rule stability across offerings to guard against drift-based "surprises".

Platforms such as the just cited KEEL (for evolutionary learning with extensive benchmarking and statistical tests, Alcalá-Fdez et al., 2009, 2011) and WEKA (for broad machine-learning workflows, scripting, and visualization, Hall et al., 2009), both widely used to prototype and compare rule learners before institutional deployment, are flanked by system like DAMBE, originally designed for molecular biology. DAMBE demonstrates a transferable design pattern: modular structure with statistically robust analytics and good visualization support (Xia & Xie, 2001). DAMBE itself is not an EDM tool. However, its architecture resembles what education teams are looking for: integrated preprocessing, exploratory analysis, model building, and interpretable output in a single, extensible environment. Tool usability and transparency – not just raw accuracy – are central to adoption; rule-based personalization systems succeed when teachers and organizations can inspect, simulate, and contest the logic that affects students and institutions.

In conclusion, these considerations on the use of evolutionary methods for rule generation in educational environment personalization identify three priorities. First, interpretability by design: prefer evolutionary rule learners that produce compact, monotone-respecting policies and ship with explanation and simulation tooling. Second, fairness-aware objectives: shift equity metrics to primary optimization targets, regularly monitor subgroup performance, and ensure human oversight for high-stakes decisions (Hardt, Price & Srebro, 2016; Slade & Prinsloo, 2013). Third, operational scalability: pair MOEAs with parallel/island implementations, adopt streaming evaluation, and package rules as versioned artifacts so policies can be rolled back or compared across terms. With these practices, evolutionary rule generation becomes not merely a research technique but a dependable engine for personalized courseware that is effective, auditable, and ethically grounded.

# Chapter 14

## Machine Learning and Evolutionary Computation

The intersection of machine learning (ML) and evolutionary computation (EC) opens new pathways to enhance educational systems (and research on them) through optimized predictive modeling, interpretable rule induction, and adaptive learning strategies. When Genetic Algorithms (GAs) and related EC methods are woven into ML pipelines, institutions are offered powerful tools for personalized learning at scale, efficient resource allocation, and equitable decision-making under real-world constraints such as limited advising capacity, shifting curricula, and heterogeneous learner populations.

In practice, hybrid ML-EC systems can help educators answer recurring, high-stakes questions with data and domain knowledge in the loop: which students need support at this moment? What is an effective intervention that is also feasible? What is the best sequence of courses or modules that balances workload? How do we keep equity while scaling? By casting such questions as search or optimization problems and then solving them with population-based heuristics that explore diverse alternatives, educational leaders can select among optimal policies in the Pareto front, trading off competing objectives (e.g., dropout reduction vs. staff effort; mastery gain vs. time-on-task), and can re-optimize as data, goals, and constraints change over time. This capacity for ongoing adaptation is essential in contemporary online environments that host MOOCs, blended programs, and workplace upskilling courses, where learner streams are dynamic, and teaching offerings change frequently.

Such comparisons resonate with ongoing debates in cognitive architectures (Langley, Laird & Rogers, 2009) about balancing symbolic and subsymbolic approaches to learning.

In this chapter, we describe the class of approaches known as Genetics-Based Machine Learning, facing applications like feature selection and rule induction, and focusing on questions related to scaling and fairness. Then, we explore the connection of evolutionary solutions to AutoML, LLM, deep learning, neural networks (and neuroevolution), showing key points, possible use cases, and limitations.

## 14.1 Genetics-Based Machine Learning (GBML)

Genetics-Based Machine Learning (GBML) is a class of approaches that couples Genetic Algorithms with machine-learning models to optimize predictive accuracy, explainability, and operational fit in dynamic environments such as educational systems (Kovacs, 2012; Fernandez et al., 2010). GBML leverages evolutionary operators (selection, crossover, mutation) and proper fitness functions to tune feature sets, hyperparameters, and model structures, and to find decision rules that are compact enough for auditing and deployment in high-stakes settings. A typical architecture is shown in Figure 14.1. This integration has two practical advantages in education: (i) adaptability when data distributions shift (e.g., curriculum changes, new assessment formats, evolving cohorts), and (ii) multi-objective control over trade-offs among learning outcomes, staff capacity, cost, and fairness (El Yessefi et al., 2024; Al Farissi et al., 2020; Nogueira et al., 2024; Deb et al., 2002; Zhang & Li, 2007).

### 14.1.1 Why GBML in education?

Classical ML methods typically assume stationarity: the mapping from inputs to outcomes is stable over time. Educational data rarely behaves that way. As instructors revise syllabi, tools get added to the platforms, semesters turn over, and external shocks (e.g., modality shifts) arrive, concept drift occurs and statistical relationships change (Gama et al., 2014). GBML addresses this by (a) re-evolution at periodic intervals (re-searching feature sets, thresholds, and rule policies each term) and (b) streaming evaluation like prequential test-then-train workflows that detect drift early and trigger re-optimization. GBML can rapidly adapt, rather than fine-tuning a single brittle model, because populations maintain diversity of candidate solutions.

In addition, education issues are multi-stakeholder and multi-objective: advisors and tutors value recall for at-risk identification; faculty require precision to limit false alarms; students appreciate transparency and agency; leadership demands efficiency and compliance. MOO via MOEAs (e.g., NSGA-II, MOEA/D) surfaces sets of policies on a Pareto front, enabling governance bodies to select options that satisfy capacity and

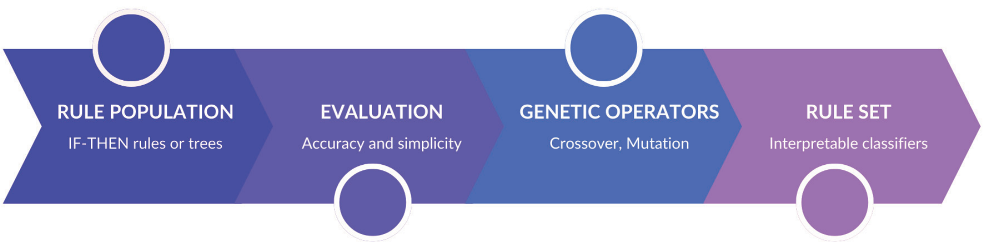


**Figure 14.1** - Genetics-Based Machine Learning (GBML) architecture

Genetics-Based Machine Learning architecture evolving rule populations under accuracy and parsimony.

equity constraints (Deb et al., 2002; Zhang & Li, 2007; Zhou et al., 2011). In practice, a chromosome might encode: (i) a sparse feature mask; (ii) interpretable rule thresholds (e.g., “<60% weekly activity + two missed quizzes → outreach in 48 h”); (iii) throttles on intervention volume; and (iv) fairness constraints (e.g., equal opportunity gaps below a threshold). Fitness is vector-valued and can, for example, include improvements in mastery, withdrawal rate, advisor hours, and subgroup parity deltas (Hardt, Price & Srebro, 2016; Zafar et al., 2017).

### 14.1.2 Feature selection and hyperparameter optimization

A central GBML pattern in Educational Data Mining (EDM) is genetic algorithm-based feature selection (GAFS). GAFS reduces noise, curbs overfitting, and lowers compute, especially when telemetry is high-dimensional (clickstreams, text, video interactions). In a representative study on student-performance prediction, Al Farissi, Dahlan, and Samsuryadi (2020) combined GAFS with Random Forests, improving accuracy while simplifying the input space (an effect consistent with broader reviews of feature selection in ML; see Pudjihartono et al., 2022). For hyperparameter optimization, GBML outperforms manual search and can be more sample-efficient than exhaustive sweeps, especially when the objective is multi-criteria (accuracy, training time, calibration, fairness).

In Figure 14.2, we show a contour view of a simulated hyperparameter response surface, motivating global search (e.g., GA) under non-convex, multi-modal conditions.

### 14.1.3 Rule induction for personalization and auditability

Education is a high-stakes, high-accountability domain. This is why interpretable rules are preferred over opaque policies when triggering outreach, altering pacing, or recommending course sequences. GBML supports rule-set learning through evolutionary algorithms such as Ant-Miner (ant-colony optimization-based rule discovery, Parpinelli, Lopes & Freitas, 2002) and genetic rule learners (e.g., GAssist, evolutionary fuzzy rule-based systems). These learners produce compact antecedent–consequent rules with explicit thresholds, which instructors can explore, simulate, and edit. Typical fitness combines predictive metrics and structural penalties for complexity, encouraging rules educators can explain to students (“Because your activity dropped below X for two weeks, we scheduled a 15-minute check-in.”). In live deployments, to reduce overfitting and bias, teams can add guardrails related to such as monotonicity (e.g., more absences shouldn’t increase mastery predictions), minimum support, and subgroup performance checks.

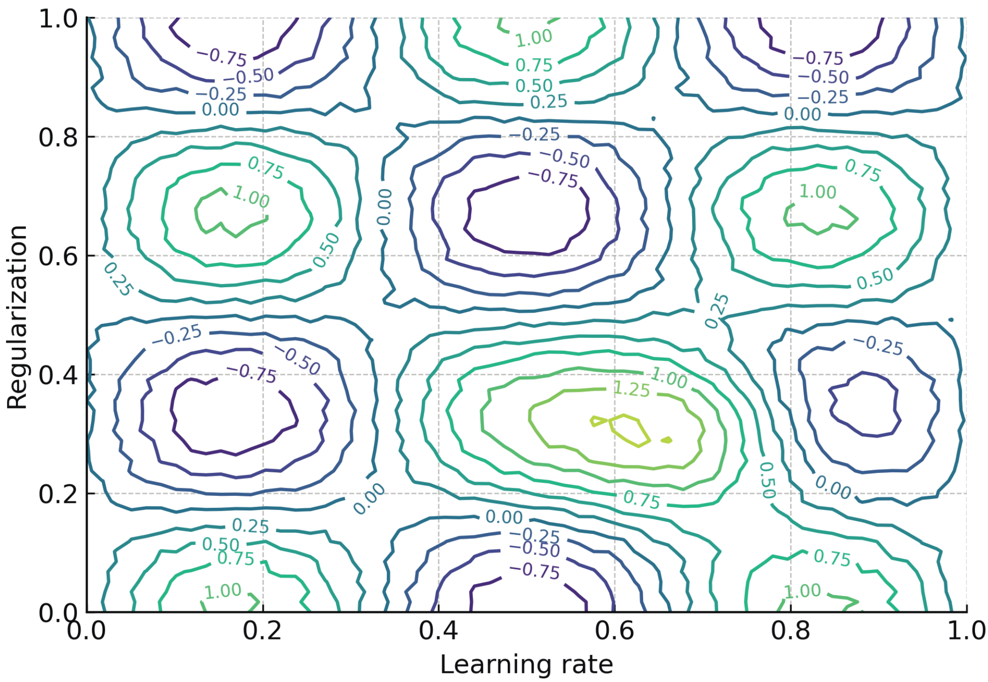


**Figure 14.2** - Hyperparameter Landscape

Contour view of a simulated hyperparameter response surface, motivating global search (e.g., GA) under non-convex, multi-modal conditions.

## 14.1.4 Hybrid GBML + neural models (neuroevolution and outer-loop search)

GBML also serves as an outer-loop optimizer for neural and tree ensembles. Historically, GAs tuned network topologies and weights (neuroevolution) for task-specific adaptation (Harp, Samad & Guha, 1989). In today's EDM practice, GAs and MOEAs select sparse, stable features, calibrate learning rates and regularization, and optimize early-warning thresholds jointly with model hyperparameters to maximize educational utility: e.g., maximize recall at fixed advisor capacity, maintain probability calibration for triage, and control alert fatigue. In immersive and adaptive environments, hybrid solutions improved context-specific predictions and responsiveness to learner signals (Bhattacharjee et al., 2018). The key is closed-loop personalization: the predictive model estimates short-term learning gains; the evolutionary layer searches policy space for sequencing, hinting, or outreach rules that achieve mastery increase per unit time under resource constraints.

### 14.1.5 Scaling GBML: standards, streaming, and parallelism

Real programs succeed on plumbing as much as algorithms. Two telemetry standards that are now widely used within higher education – Experience API (xAPI) and Caliper Analytics – simplify the operationalization of GBML pipelines by standardizing the event vocabularies and payloads used when logging events within educational applications (Advanced Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; 1EdTech Consortium, 2020). Standardized traces allow for (i) maintaining consistent features across courses and terms, (ii) performing prequential (streaming) evaluation to detect drift, and (iii) scheduling re-evaluation runs on a cadence (weekly, termly) or on anomaly triggers (Gama et al., 2014). GBML exploits parallelism and the island model for computation, testing candidate solutions in parallel and periodically swapping elites. These practices are popular in open toolchains (e.g., KEEL; WEKA add-ons) and cloud clusters (Alcalá-Fdez et al., 2009, 2011; Hall et al., 2009).

### 14.1.6 Fairness, privacy, and human-in-the-loop governance

GBML design must consider equity and ethics as first-class constraints. Fairness measures such as equal opportunity (equal true-positive rates across groups) and variants are easy to integrate within the objectives or constraints of MOEAs, leading to policies that give up a small amount of performance for large gains in fairness (Hardt, Price & Srebro, 2016; Zafar et al., 2017). Privacy-by-design principles – data minimization (via GAFS), role-based access, and transparency – align with sector guidance such as DELICATE (Drachsler & Greller, 2016) and broader LA ethics frameworks (Slade & Prinsloo, 2013). Finally, effective programs are human-in-the-loop: tutors can reject recommendations, providing justification for their decision; committees approve policy changes; and students are given explanations and recourse. Teachers and tutors remain accountable; GBML is a decision support system, not an automated judge.

### 14.1.7 From prototype to practice: toolchains and evaluation

KEEL provides a repository of evolutionary learners (including Ant-Miner variants, genetic rule learners, and evolutionary fuzzy systems), preprocessing methods, and nonparametric statistical tests for prototyping and comparing algorithms (Alcalá-Fdez et al., 2009; 2011). WEKA, a well-known Data Mining Software, complements this with a broad model zoo and scripting for pipelines (Hall et al., 2009). Public datasets such as OULAD (Open University Learning Analytics Dataset) facilitate reproducible baselines and ablation studies prior to institutional pilots (Kuzilek, Hlosta & Zdrahal, 2017; Hlosta et al., 2018). In production, teams prefer sequential testing (A/B or stepped-wedge), calibration checks (e.g., reliability diagrams), and operational metrics (advisor workload, alert fatigue), alongside learning outcomes. Results are versioned, and policy artifacts (rule sets, thresholds, masks) are stored so that changes are auditable and rollback is possible (practices consistent with MLOps norms and educational governance).

## 14.2 AutoML with evolutionary search, LLM-augmented GBML for text-rich signals, and MLOps patterns for academic institutions

Educational programs increasingly need end-to-end pipelines that are accurate but also operable: easy to maintain, auditable, fair, and resistant to drift. This section brings together three complementary strands that make that possible at scale and may serve as a basis for future development in educational research:

- *AutoML with evolutionary search*, to automate feature selection, model/hyper-parameter tuning, and even pipeline design under multiple, competing objectives (e.g., accuracy, advisor capacity, and latency). See Figure 14.3 for an example;
- *LLM-augmented GBML for text-rich educational signals*, to turn messy, high-volume language data (forum posts, reflections, feedback, tickets) into structured features, labels, and rules;
- *MLOps patterns tuned to universities*, to productionize the above with governance, telemetry standards, continual evaluation, privacy/fairness controls, and human-in-the-loop decision support.

Together, these patterns let institutions evolve policies (not just models): who to support, in what ways, and when, subject to capacity and equity constraints, then re-optimize as context shifts.

### 14.2.1 AutoML via evolutionary search: from pipelines to policies

Why evolution for AutoML? Educational data may be stored in heterogeneous silos (LMS logs, advising notes, Student Information System records, forums, proctoring events). Model performance hinges on the right combination of preprocessing, featurizers, estimators, and thresholds, as well as on operational constraints (compute budgets, inference latency, fairness, and capacity limits). Casting this as a combina-

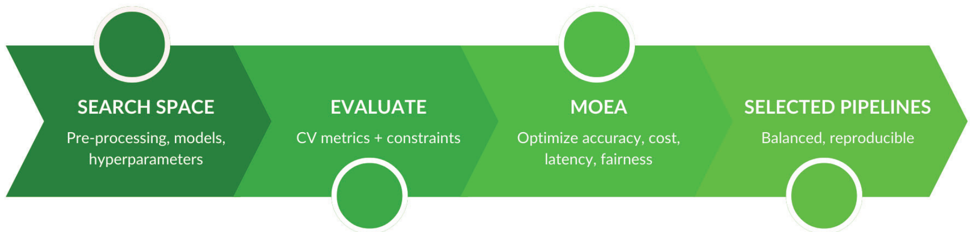


**Figure 14.3** - An AutoML pipeline selection

AutoML with EC searching over preprocessing, models, and hyperparameters under multiobjective criteria.

torial search problem is natural; evolutionary algorithms explore that space efficiently while preserving diversity and surfacing multiple Pareto-optimal candidates.

Some key considerations.

*Pipelines as chromosomes*. In genetic programming-style AutoML, a pipeline is an expression tree (e.g., missing-value imputer → scaler → feature selector → classifier). TPOT is a canonical example: it uses GP to evolve scikit-learn pipelines end-to-end, optimizing cross-validated performance while penalizing complexity (Olson et al., 2016). For educational teams, this means the system can discover that a sparse logistic model with GA-selected features and calibrated probabilities outperforms a heavier ensemble for near-term early-warning, and fits the latency budget for a weekly advising workflow.

*Neural architectures and resource costs*. For deep models (e.g., text or video), neural architecture search (NAS) via evolutionary strategies has proven competitive with reinforcement learning. Regularized evolution (aging evolution) found image classifiers that matched or exceeded RL-NAS while being simpler to implement and parallelize (Real et al., 2019). In practice, program teams use MOEAs to trade off accuracy with model size, inference cost, and explainability, producing compact architectures that can run within LMS or analytics dashboards without GPU dependencies. Benchmark initiatives such as NAS-Bench-101 provide standardized, reproducible environments for comparing neural architecture search algorithms, avoiding unfair evaluation of evolutionary and reinforcement learning approaches (Ying et al., 2019).

*Multi-objective AutoML*. Educational deployment is a compromise between accuracy and operational fit. Using NSGA-II or MOEA/D, institutions can evolve pipeline populations under vector-valued fitness based on: (i) AUC/PR for at-risk detection, (ii) expected weekly intervention volume (advisor capacity), (iii) fairness deltas (e.g., equal opportunity across subgroups), and (iv) latency/cost targets for nightly batch jobs (Deb et al., 2002; Zhang & Li, 2007; Zhou et al., 2011; Hardt, Price & Srebro, 2016; Zafar et al., 2017). The result is an array of non-dominated pipelines for leadership to review and approve with documented trade-offs.

*Practical tips*. Adopt island models or distributed evaluation to ensure scalability, combined with early-stopping and caching to prevent retraining identical subtrees, and warm-start populations with domain-reasonable baselines (e.g., sparse linear models) to speed convergence. For auditability, log all artifacts, including code, parameters, seeds, and data versions. Finally, reserve a final, frozen hold-out for decision-making (not just CV) before rollout.

### 14.2.2 LLM-augmented GBML for text-rich educational signals

Educational ecosystems generate language at scale: discussion forums, reflective journals, open-response items, peer feedback, office-hours notes, support tickets, and even code-review comments. Usually, this text was underutilized due to the costs of labeling and feature engineering. Large Language Models (LLMs) have changed that, first by producing high-quality embeddings that capture semantics in compact vectors, and second by serving as versatile labeling and extraction engines when combined with evolutionary outer loops.

We now turn to some issues to keep in mind.

*From text to features*. Off-the-shelf general-purpose encoders such as BERT and Sentence-BERT (Devlin et al., 2019; Reimers & Gurevych, 2019) provide sentence- and document-level embeddings that correlate with discourse coherence, help-seeking attitudes, and affective signals predictive of outcomes. GBML subsequently selects features from these embeddings (or those concatenated with clickstream features), producing sparse, stable subsets that preserve performance while economizing on compute and storage, which is a core requirement for compliance and privacy minimization (Al Farissi, Dahlan & Samsuryadi, 2020).

*Prompt and label optimization with evolutionary loops*. LLMs can be guided via prompts to label small corpora (e.g., "struggling", "off-topic", "needs resources"). But prompt quality varies. Two recent lines of work, Promptbreeder and EvoPrompt, use optimization (gradient-free search and evolutionary operators) to automatically improve prompts, closing the loop between prompt variants and and performance on the task (Fernando et al., 2023; Guo et al., 2024). This can be supplemented with manual spot-checks and KEEL/WEKA comparisons to ground truth the labels before GBML rule induction (Alcalá-Fdez et al., 2009; Hall et al., 2009).

*Rule induction for explainability*. Once text embeddings and labels have been established, evolutionary rule learners (e.g., Ant-Miner; evolutionary fuzzy rule-based systems) can generate interpretable triggers for action: "If (help-seeking discourse ↓) AND (weekly activity ↓) THEN send scaffolding resources within 48h." These rules can also include guardrails (minimum support, subgroup parity checks) and be subjected to review by faculty committees prior to deployment, aligning with ethical frameworks in Learning Analytics (Parpinelli, Lopes & Freitas, 2002; Drachsler & Greller, 2016; Slade & Prinsloo, 2013; Hardt, Price & Srebro, 2016).

*Generative assistance with guardrails*. LLMs can also synthesize personalized nudges (e.g., feedback referring to each student's goals). In these cases, GBML might optimize policy parameters – optimal time-to-send, frequency caps, tone templates – against outcomes like mastery gain, opt-out rate, and teacher / tutor workload. Institutions should pair generation with Model Cards (Mitchell et al., 2019) and Datasheets (Gebru et al., 2021) to document purpose, data sources, and known limitations, plus human override and opt-out affordances for students.

*Privacy-preserving setups*. Sensitive text should be minimized, anonymized, and processed via federated or on-prem options where possible; where centralization is necessary, add differential privacy mechanisms and access controls. Evolved feature masks often help by retaining only a few predictive, non-identifying aggregates (Kairouz et al., 2021; Dwork & Roth, 2014; Drachsler & Greller, 2016).

### 14.2.3 MLOps patterns for academic institutions

The discipline around data standards, versioning, testing, monitoring and governance is necessary to advance from prototyping to production, but it must be made within the constraints of education: semester cadence, academic committees and regulations such as those from FERPA (Family Educational Rights and Privacy Act), GDPR (General Data Protection Regulation), or IRB (Institutional Review Board).

Below are some practical recommendations for MLOps (Machine Learning Operations), which refers to the set of practices that manage the entire lifecycle of machine learning models.

*Telemetry and schemas*. Implement xAPI and Caliper to share standardized event vocabularies for assignments, media interactions, or forum posts. This will make features portable across courses/programs enabling prequential evaluation and reproducible GBML runs (Advanced Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; 1EdTech Consortium, 2020).

*Version everything*. Version datasets (and their Datasheets), models (and their Model Cards), and policies (rule sets, thresholds). Use a lifecycle tool (e.g., MLflow or an equivalent) to log runs, parameters, metrics, and artifacts so changes are auditable and rollback is possible (Mitchell et al., 2019; Gebru et al., 2021).

*Continual evaluation and drift*. Rely on prequential (test-then-train) evaluation for data streams; monitor calibration and subgroup performance; define P-stops (policy stops) that automatically disable a model if fairness gaps or calibration drift exceed thresholds. If drift, trigger re-evolution of masks, hyperparameters, and rules (Gama et al., 2014; Hardt, Price & Srebro, 2016; Zafar et al., 2017).

*Human-in-the-loop governance*. GBML outputs should be advice, not immutable decisions. Tutors can override with rationale logging; committees approve new rules after reviewing trade-off plots in the Pareto fronts. Follow DELICATE principles (Determine, Explain, Legitimate, Involve, Consent, Anonymise, Technical, Externalize) to ensure ethical use of analytics (Drachsler & Greller, 2016; Slade & Prinsloo, 2013).

*Risk management and compliance*. Align with NIST AI RMF 1.0 for risk identification, measurement, and mitigation; document intended use, out-of-scope cases, and known failure modes. When using LLMs for generation, apply filters and checklists for harmful or biased content before sending communications to students (NIST, 2023; Mitchell et al., 2019; Gebru et al., 2021).

*Right-sizing for smaller programs*. Not every unit needs a cluster or large-scale computational infrastructures. Practical pattern: start with TPOT (or DEAP-based scripts) on curated features; wrap with weekly batch scoring; review dashboards in program meetings; iterate every term. As needs grow, add MOEA-based policy optimization and streaming evaluation (Fortin et al., 2012; Olson et al., 2016; Ifenthaler & Yau, 2020).

### 14.2.4 Illustrative vignettes (design patterns in practice)

*Vignette A: early-warning AutoML with capacity-aware policies*

Goal: to deliver timely, interpretable, and human-governed feedback based on a combination of text-derived signals and behavioral data while maintaining responsibility from faculty and keeping documentation.

A teaching & learning center uses TPOT to evolve pipelines on the OULAD dataset (and its local equivalent), optimizing AUC with a fairness penalty and a hard constraint on weekly outreach capacity (e.g., 400 calls per week). NSGA-II surfaces a Pareto set that includes a sparse logistic model (fast, transparent) and a small gradient-boosting model (higher AUC but more complex). The committee selects the sparse model for first-term deployment, documents trade-offs in a Model Card, and sets a P-stop when the prevalence of opportunity gaps is greater than 5 percentage points (Kuzilek, Hlosta & Zdrahal, 2017; Olson et al., 2016; Deb et al., 2002; Hardt, Price & Srebro, 2016; Mitchell et al., 2019).

*Vignette B: LLM-augmented discourse signals with interpretable triggers*

Goal: to deliver timely, interpretable, and human-governed feedback by combining text-derived signals and behavioral data, while preserving faculty responsibility and documentation.

An online program encodes weekly forum posts with Sentence-BERT, then uses a small, LLM-assisted labeling round to classify posts into help-seeking, struggle, or progress. An evolutionary rule learner (Ant-Miner) induces interpretable rules combining discourse labels and clickstream features to trigger short, personalized nudge templates authored by faculty. Prompts for LLM-generated feedback are tuned with an evolutionary prompt search loop (EvoPrompt/OPRO), but final messages remain faculty-approved and rate-limited to protect student autonomy. Datasheets/Model Cards document the scope and opt-out options (Reimers & Gurevych, 2019; Parpinelli, Lopes & Freitas, 2002; Yang et al., 2024; Guo et al., 2024; Gebru et al., 2021; Mitchell et al., 2019).

*Vignette C: Curriculum MOEA for adult learners*

Goal: to identify and validate curriculum sequences that maximize effective learning while respecting time constraints and participants' experience, supporting informed rather than automated teaching decisions.

A workforce program evolves alternative module sequences using NSGA-II with objectives: mastery probability (from a calibrated model), weekly time budget, and satisfaction proxies. The Pareto front presents "intensive onboarding then steady pace" vs. "gradual ramp with more formative practice". Faculty choose two to A/B test; results are logged and feed the next MOEA cycle (Nogueira et al., 2024; Deb et al., 2002).

### 14.2.5 Risks and mitigations

- Label leakage and spurious correlation: evolutionary search can find solutions that key off artifacts (e.g., future information). Use leakage audits, temporal CV, and ablations. Favor stable feature masks across terms.
- Fairness regressions: include parity metrics in the objective, not just in post-hoc dashboards. Monitor subgroup calibration and opportunity gaps continuously; set P-stops (Hardt, Price & Srebro, 2016; Zafar et al., 2017).
- Drift and brittleness: prequential evaluation and scheduled re-evolution mitigate drift; use diversity preservation in populations and retain prior bests as warm starts (Gama et al., 2014).
- Privacy and governance: minimize sensitive data via feature selection; consider federated setups and differential privacy; adhere to DELICATE, Datasheets, and Model Cards (Dwork & Roth, 2014; Drachsler & Greller, 2016; Gebru et al., 2021; Mitchell et al., 2019; Kairouz et al., 2021).

## 14.3 Hybrid models combining deep learning and GAs

Combining the positive traits of deep learning with Genetic Algorithms (GAs), hybrid models can unite two complementary strengths: the first, deep nets, extract high-level representations from complex, rich, noisy data; the second, evolutionary search, explores large, irregular design spaces, overcoming poor local optima. Applying this synthesis in the context of Educational Data Mining (EDM) and Learning Analytics (LA) – where data are multi-modal (clickstreams, video interactions, text, assessments), non-stationary (subject to concept drift across terms or cohorts), and the optimization problem itself is naturally a multi-objective optimization problem (accuracy, timeliness, interpretability, compute/latency and fairness) – can be particularly powerful. In the following pages, we unpack what GAs optimize inside deep models, design patterns that have proven robust, and how these hybrids are being used to predict risk, personalize pathways, and scale support while outlining guardrails for efficiency, transparency, and equity.

What do GAs optimize in deep learning? In practice, GAs slot into three layers of a deep-learning pipeline:

- *hyperparameters and training schedules*. GAs (and closely related evolutionary strategies) search over learning rates, optimizers, batch sizes, dropout, data-augmentation policies, and training schedules. Population-based training (PBT, Jaderberg et al., 2017) famously mutates and selects whole training "lineages" online, yielding strong results with modest orchestration overhead – an approach now widely used in many large-scale pipelines.

- *architectures*. Neuroevolution methods (e.g., NEAT and its successors) evolve both topology and weights, adding nodes and connections over time. More recent "regularized evolution" and genetic-programming solutions can be targeted at modern convolutional and hybrid blocks in constrained search spaces, often matching hand-designed baselines with less manual tuning. Such approaches are appealing in education, as they can aim for lightweight models that satisfy institutional latency and hardware constraints, e.g., in browsers and entry-level laptops.
- *objective trade-offs*. As stated many times, also in this case, multi-objective evolutionary algorithms (MOEAs), including NSGA-II and MOEA/D, preserve Pareto fronts across multiple competing goals (e.g., early-warning AUC vs. explanation fidelity vs. inference cost), allowing stakeholders to select points that also align with policy & resource realities. This is important when models must be actionable and predictive for both advisors and instructors.

Below is a list of hybrid situations in which the combined use of deep learning and GA generates, or could generate, research, topics, and practices to take into account in education.

*Design patterns that work*. A recurring pattern in successful hybrids is a two-loop system: an inner loop that trains or fine-tunes a deep model on educational tasks (risk prediction, recommendation, pathway sequencing), and an outer GA loop that proposes new configurations (hyperparameters, augmentations, modules) and selects based on validation performance and additional constraints (fairness, latency, cost). To keep search tractable, teams leverage (i) weight sharing or one-shot supernets, (ii) pre-computed NAS benchmarks, and (iii) warm-starting from prior cohorts or courses. These tactics cut down the GPU budget without losing search quality.

*Early-warning and retention models at scale*. Deep networks have already performed successfully with clickstream sequences and assessment traces in virtual learning environments (VLEs); evolutionary wrappers further improve robustness and operational fit. For instance, deep recurrent/attention models on VLE big data can be paired with GA-based feature selection to reduce input dimensionality, stabilize performance across cohorts, and keep inference costs low (important for rolling, week-by-week risk scoring). Evidence from large deployments shows deep models alone can predict outcomes effectively; GA-enabled selection and scheduling help make those models lighter and more maintainable in production.

*MOOCs and adaptive pathways*. In the context of Massive Open Online Courses, hybrid models can merge temporal modeling (LSTM/Transformer blocks) with GA-driven hyperparameter and curriculum policy search to perform: (a) forecasting disengagement windows; (b) tailoring interventions (recommendations, practice sets, alternate explanations) before attrition peaks. Model-driven, real-time recommendations in MOOC performance prediction can improve attainment; evolutionary outer loops make such systems portable across offerings by re-tuning to new content mixes, calendars, and student demographics.

*Curriculum learning and schedule optimization*. Deep models often learn more effectively when exposed to training examples in a carefully staged order (curriculum learning). Evolutionary search, which can optimize these curricula online by selecting near-optimal training environments/schedules using simple local rules, could significantly reduce engineering overhead and increase sample efficiency. The same idea in educational sequencing holds: with constraints such as time-on-task and the satisfaction of prerequisites, a GA can evolve both the order and the granularity of the learning objects used by a deep recommender.

*Fairness, auditability, and multi-objective selection*. Educational deployments must balance accuracy with procedural justice. In MOEAs, it is possible to add fairness regularizers (equalized odds gaps and false-negative rate disparities) as components of the fitness function, in addition to AUC and latency. This results in a Pareto set of auditable models from which decision-makers can pick a solution that, at the same time, meets performance and equity targets, aligning with established fairness formalisms. This choice should be associated with subgroup evaluation and error budgeting to avoid "fairness gerrymandering".

*Search spaces beyond knobs: data pipelines and interventions*. In education, what you optimize often matters more than the last 1-2% of predictive lift. Hybrid GA+DL systems can evolve not only model hyperparameters but also data-pipeline steps (temporal windows, resampling strategies, missing-data handling) and intervention policies (who to contact, when, and how) under resource constraints. This aligns with "whole-system" optimization, evolving models and workflows together to reduce advisor alert fatigue and improve student outcomes.

*Efficiency tactics for real institutions*. To keep compute budgets realistic, teams use a combination of (a) evolution-inspired online training (PBT); (b) compact, hardware-aware architectures discovered by evolution; (c) weight-sharing NAS; and (d) Covariance-Matrix Adaptation (CMA-ES) for continuous hyperparameters. These mitigate wall-clock time and energy while maintaining accuracy, which is essential for institutions with limited infrastructure.

*Interpretability and human-in-the-loop oversight*. Deep models can be paired with local explanation tools (e.g., LIME, SHAP) to expose the factors behind an alert for a given student. Evolution then runs in the outer loop to favor configurations that are both high-performing and explainable, e.g., by penalizing instability in explanations across retrains or maximizing sparsity in attention patterns. This stimulates advisors and instructors to adopt and use them responsibly.

*Concept drift and continuous adaptation*. Educational data drift across terms (new cohorts, changed syllabi, platform updates). Evolutionary outer loops lend themselves to continual re-tuning: periodically evolving hyperparameters, augmentations, or even sub-architectures to re-fit the current distribution while enforcing stability constraints (e.g., bounded change in calibration). Drift-aware evaluation protocols – rolling-window validation and prequential testing – should be standard.

## Putting it together: an implementation blueprint

A practical hybrid stack for early-warning and pathway recommendation might look like this:

1. *Data layer*: sessionized clickstream, LMS events, assessment logs, demographics, and calendar signals; basic quality checks and leakage guards.
2. *Model inner loop:* a sequence model (Transformer/GRU) plus a small tabular MLP head; optional text/video encoders if content signals are available.
3. *Evolutionary outer loop:* a GA searches across hyperparameters (optimizer, LR schedule), data-window parameters, class-imbalance tactics, and lightweight architectural choices (number of heads/layers). The fitness combines AUC/PR at weekly horizons, calibration, training cost, latency, and fairness gaps.
4. *Multi-objective selection:* NSGA-II or MOEA/D returns a Pareto front; stakeholders select operating points given policy and capacity.
5. *Interpretability & governance:* SHAP/LIME on final candidates; stability checks across resamples; human-in-the-loop review of alert thresholds before rollout.
6. *Online adaptation:* PBT-style mutations on living models; drift monitors trigger limited evolutionary re-search when performance decays.

Education-specific use proposals follow.

- *Video learning analytics with adaptive remediation.* Fine-grained video interaction signals (pause/seek patterns) are “read” by deep models to mark confusing segments. Next, a GA searches for which augmentation and temporal pooling strategies best match predictions to human ratings of difficulty, thereby improving recommendations for “micro-remediation” clips without blowing up the compute.
- *Program-level at-risk identification with lightweight inference.* A recurrent model trained on semester-level sequences, wrapped by a GA feature selector and CMA-ES LR tuning, yields fast, stable alerts suitable for weekly advising cycles even on shared servers. In large VLE deployments deep models can reach production readiness; evolution adds robustness and cost control.
- *Curriculum scheduling and practice sequencing.* Evolutionary curriculum schedulers can also simplify the design of policies, in agents training using reinforcement-learning methods or in adaptive practice systems: evolve easy-to-implement rules that make convergence easier, and student practice more efficient with fewer knobs to set manually.

Limitations and safeguards: hybrids increase algorithmic complexity and the risk of silent failures (e.g., reward hacking in the outer loop, data leakage in the inner loop). Mitigations include: strict train/validation splits that respect temporal order; reproducible seeds and search logs; NAS benchmarks to dry-run search strategies; and fairness-aware fitness components with subgroup reporting. Institutions should also monitor environmental and financial costs, favoring weight sharing, early stopping, and hardware-aware search to minimize GPU hours.

## 14.4 Evolving neural networks and adaptive systems

The optimization of neural networks with genetic and evolutionary algorithms – often referred to as neuroevolution – has evolved into a practical and increasingly adopted set of techniques for building adaptive analytics and decision-support systems, including emerging applications in education. Where deep learning serves as a powerful class of function approximators that enables highly expressive multimodal learning traces (e.g., clickstreams, assessments, text, and video), evolutionary search facilitates global exploration with increased robustness to non-convex optimization landscapes and multi-objective control over auxiliary criteria such as latency, stability, and fairness. The resulting hybrid systems can be adapted to local constraints (hardware, institutional policies, pedagogical requirements) and, when appropriately designed, can be updated over time to accommodate evolving cohorts, curricula, and platforms.

### 14.4.1 Core evolutionary operators for neural optimization

At the heart of GAs are mutation and crossover (Mitchell, 1995). Mutation introduces diversity by perturbing architectural motifs (e.g., layer widths, activation choices) or continuous parameters (e.g., learning rates, weight decay), which helps escape premature convergence and track concept drift (Gama et al., 2014) across terms or cohorts. Crossover recombines “good parts” of parent solutions, accelerating the discovery of high-performing configurations without exhaustive grid search. These operators are well-suited to educational data, where behavior can be non-stationary (new content versions, policy changes, calendar shifts) and objectives may conflict (e.g., maximize early-warning recall while minimizing false alarms and inference cost).

In contrast to purely gradient-based hyperparameter tuning, evolutionary search maintains a population of candidate models/configurations trained in parallel; selection keeps promising lineages, while mutation and crossover explore. Already cited popular variants include Covariance Matrix Adaptation (CMA-ES) for continuous hyperparameters (Hansen & Ostermeier, 2001) and Population-Based Training (Jaderberg et al., 2017), which online mutates learning rates, augmentations, or optimizer choices as training proceeds, particularly attractive when one wants to adapt models during a semester without restarting training from scratch.

### 14.4.2 What we evolve: weights, topologies, curricula, and policies

Neuroevolution spans a spectrum (Stanley & Miikkulainen, 2002; Elsken, Metzen & Hutter, 2019):

- *weight evolution:* evolving full weight vectors is feasible for small models or final layers to meet edge-inference constraints (e.g., dashboards that must run on commodity hardware).

- *topology/architecture evolution:* methods such as NEAT (NeuroEvolution of Augmenting Topologies, Stanley & Miikkulainen, 2002) evolve both structure and weights with speciation (niching) to preserve innovation; modern regularized (or aging) evolution and genetic programming (GP) discover compact convolutional or hybrid blocks under tight compute budgets (Real et al., 2019; Suganuma, Shirakawa & Nagao, 2017) – useful when institutions need low-latency models in browsers or LMS plug-ins.
- *indirect encodings:* encodings like CPPNs/HyperNEAT (Stanley, D'Ambrosio & Gauci, 2009) produce large, regular networks from compact genotypes – handy when scaling architectures while maintaining parameter efficiency.
- *training schedules and curricula:* evolution can select data orders, augmentation policies, or curriculum learning schedules (Jiwatode, Schlecht & Dockhorn, 2024) that improve sample efficiency and stability – critical for small cohorts or rapidly changing content.

Crucially, in educational settings, we often optimize beyond accuracy. Multi-objective evolutionary algorithms (MOEAs) such as NSGA-II and MOEA/D return Pareto fronts over AUC/PR, calibration, explanation stability, fairness gaps, and inference time, allowing deans or analytics leads to pick solutions that fit governance and resource realities (Deb et al., 2002; Zhang & Li, 2007).

### 14.4.3 From models to adaptive systems

When we approach education procedures and research, evolving a network is only part of the story. For practice to change, models need to be embedded in adaptive systems that personalize pathways, schedule group work, allocate faculty time, or trigger timely interventions. GAs can fit naturally as outer-loop optimizers around deep models in the following systems:

1. *early-warning and retention.* Deep sequence models for VLE data (clicks, submissions, forum actions) represent robust baselines for outcome prediction. A GA outer loop can jointly evolve (i) input windows and resampling strategies, (ii) hyperparameters/architectures, and (iii) alert thresholds to strike a balance between recall and teachers' workload. Evidence from VLE research shows that deep models predict outcomes effectively; evolutionary wrappers help reduce input dimension, stabilize performance across cohorts, and respect inference budgets for weekly runs (Waheed et al., 2020; Sahlaoui et al., 2024; Queiroga et al., 2020; Charitopoulos, Rangoussi & Koulouriotis, 2020).
2. *MOOCs and micro-interventions.* In Massive Open Online Courses, hybrid GA+DL systems predict disengagement windows and then optimize intervention policy under resource limits (How many nudges per week? Who gets what content?). MOEA-based selection lets designers trade off completion rates, timeliness, and CPU/GPU consumption, while keeping explanations consistent for advisors.

Complementary work on MOOC performance prediction shows that model-driven recommendations can improve attainment, and evolutionary search helps retune these systems as course mixes and calendars change (Hao, Gan & Zhu, 2022; Zakaria, Anas & Oucamah, 2022).

3. *adaptive grouping in collaborative learning*. GA-based grouping, deployed in university courses, has produced higher-quality results and more balanced participation, allowing instructors to adjust constraints on roles, skills, and schedules (Li, Ouyang & Chen, 2022). Web tools (Ugarte et al., 2022) used GA fitness to balance intra-group heterogeneity (for peer learning) and inter-group comparability (for fair assessment). Research reported that GA-based solutions outperform heuristic or random grouping in terms of satisfaction and output quality (Moreno, Ovalle & Vicari, 2012). By leveraging deep models (e.g., embeddings derived from interaction traces), the features exploited by GAs might be further enriched, enabling groupings that change during a course over time as new evidence comes in.
4. *sequencing and curriculum optimization.* In adaptive e-Learning, GAs evolve content sequences (Hssina & Erritali, 2019) with penalties for overload and redundancy. Case studies show GA planners converge toward coherent, personalized module orders with lightweight data inputs, practical for institutional pilots. In other research, evolutionary curriculum schedulers have been shown to yield performance improvements during the training of a reinforcement learning agent (Jiwatode, Schlecht & Dockhorn, 2024). Deep models might estimate learning situations and predicted gains from candidate items.
5. *timetabling and resource allocation with learning in the loop.* GA timetabling with preferences can be linked to predictive models of classroom utilization and bottlenecks, allowing institutions to trial "what-if" schedules versus projected student burden (stress or clash load). GA timetabling studies, which include some form of student preference (Dunke & Nickel, 2023; Kučak, Juričić & Đambić, 2019; Mahlous & Mahlous, 2023) have demonstrated feasibility and greater satisfaction.

### 14.4.4 Making neuroevolution practical in education

*Compute budgets and scalability*. Evolution has a reputation for expense, but several tactics make it tractable for universities:

- weight sharing & tabular NAS baselines: share parameters across candidate subnets and use small, reproducible search spaces to keep GPU hours low (Elsken, Metzen & Hutter, 2019).
- regularized evolution with small populations and early stopping yields hardware-efficient models that are competitive with hand-designed models (Real et al., 2019).
- PBT allows training and evolution to coexist, reusing compute already spent on SGD (Jaderberg et al., 2017).
- CMA-ES tunes continuous hyperparameters (e.g., label-smoothing or mixup strength) in very few evaluations (Hansen & Ostermeier, 2001).

*Robustness to drift*. Educational data changes from semester to semester. Implementation strategies, including rolling-window validation, prequential testing, and periodic re-evolution of a narrow slice of the configuration (e.g., only data windows & thresholds) that maintain calibration and subgroup performance without full retrains.

*Interpretability and governance*. Adoption hinges on explainability. Even when the predictor is deep, the evolutionary outer loop can favor candidates with stable SHAP value patterns or with sparse attention maps, and penalize solutions that exhibit explanation volatility across folds. When coupled with LIME/SHAP in the advisor dashboards, this preserves trust while encouraging actionably specific advice (Ribeiro, Singh & Guestrin, 2016; Lundberg & Lee, 2017).

*Fairness as a first-class objective*. Institutions rightly worry about disparate error rates across demographic groups. For this reason, fairness terms can be added directly to the fitness functions, pushing the GA to discover models and thresholds that meet equity targets. MOEAs then show the stakeholders a Pareto front, trading small accuracy losses for large disparity reductions, which aligns with traditional fairness formalism (Hardt, Price & Srebro, 2016; Zafar et al., 2017; Deb et al., 2002; Zhang & Li, 2007).

*Data and benchmarks*. Public resources such as the Open University Learning Analytics Dataset (OULAD) allow teams to prototype neuroevolution strategies under realistic constraints and compare drift-robustness or fairness trade-offs before touching institutional data (Kuzilek, Hlosta & Zdrahal, 2017; Hlosta et al., 2018). For video-heavy courses, interaction analyses (pause/seek/replay) offer rich signals that deep models can ingest; evolution then optimizes pooling and augmentation to align predictions with human-judged difficulty (Hasan et al., 2020).

### 14.4.5 Possible applications in adaptive systems

*Pattern A*: early-warning with multi-objective selection. Start with a compact Transformer/GRU fed by weekly sessionized traces; include a small tabular head for grades as GPA and attendance. An outer GA evolves (i) look-back windows, (ii) imbalance tactics, (iii) LR/optimizer/schedule, and (iv) alert thresholds. Fitness blends AUC@k weeks, calibration error, advisor-workload penalty, and disparity metrics. NSGA-II returns a Pareto set; leadership selects an operating point, and advisors get SHAP-based explanations. Drift monitors trigger a restricted re-evolution if performance decays.

*Pattern B*: group formation with evolving constraints. Instructor-facing UIs expose weights on heterogeneity (skills, schedules), role balance, and group size. The GA proposes multiple feasible rosters with rationales; instructors can lock or swap members, after which the GA re-optimizes. A deep embedding of learner interaction/skills augments features.

*Pattern C*: sequencing and cognitive load control. A deep model related to students' mastery predicts knowledge gains and load; a GA assembles content sequences while penalizing redundancy or overload. GA planners converge to coherent, personalized module orders with minimal instrumentation.

*Pattern D*: timetabling with learner-centric objectives. Preference-aware GA timetabling incorporates student availability and commute/time-zone constraints (for blended/online), co-optimized with instructor load. This reduces conflict and improves satisfaction; when paired with predictive models of stress or risk of passing the course, the scheduler can explicitly minimize predicted overload in critical weeks.

### 14.4.6 Limitations and open directions

Despite their versatility, evolving neural networks introduce system complexity. Risks include reward hacking in the outer loop, leakage from future information, or brittle solutions exploiting accidental quirks of a term's data. Mitigations include: leakage audits; temporal splits; reproducible seeds and run logs; and search-space discipline informed by NAS surveys and benchmarks (Elsken, Metzen & Hutter, 2019). Another challenge is cost, as evolution can require significant computing power. Weight sharing, early stopping, small populations, and warm starts from prior terms meaningfully reduce costs (Real et al., 2019).

Finally, for most institutions, explainability and equity in the processes are non-negotiables. Here, the features of MOEAs with explicit trade-offs are an advantage, which shifts responsibility to governance: stakeholders must pick and justify operating points, then track them over time. When implemented correctly, evolutionary neural networks and adaptive systems not only improve predictive performance but also generate tools that adapt more quickly, operate more easily within the imposed constraints, and ensure fairer conditions for students.

# Chapter 15

# Challenges and ethical considerations

Navigating the landscape of innovation and research in education with Genetic Algorithms (GAs) and Evolutionary Computation (EC) means contending with practical challenges and ethical dilemmas that emerge when these techniques step outside the laboratory or safe simulation into live, high-stakes contexts. Scalability and computational efficiency, data governance and privacy, and the wider implications of algorithmic decision-making are not side notes but rather inform the inclusivity, equity, and sustainability of the systems. Treating these constraints as first-class design goals (not afterthoughts), institutions can delineate a responsible path in which algorithmic advances measurably support learning while remaining transparent, fair, and feasible to operate (European Union, 2016, 2024; National Institute of Standards and Technology (NIST), 2023; European Data Protection Board (EDPB), 2024).

This section provides an in-depth and structured exploration of this topic, which was introduced several times in previous sections of this volume.

In Figure 15.1, we illustrate a conceptual framework to help balance performance, fairness, privacy, transparency, accountability, and sustainability for educational optimization.

## 15.1 Scalability and computational efficiency

Educational platforms and open databases (Kuzilek, Hlosta & Zdrahal, 2017), both related to a single university or a network of them, as in MOOCs in international or regional environments (Ruipérez-Valiente et al., 2022), routinely log millions of interactions and assessment events by thousands of learners. As a consequence, the volume and heterogeneity of data, and the compute you can afford, are hard constraints that institutions meet when deploying GAs and EC within real institutional pipelines for early-warning systems, dashboards, preference-aware timetabling, grouping, or sequence optimization. Deep models trained on these streams can be highly accurate (Waheed et al., 2020), but when wrapped in an evolutionary outer loop evaluating

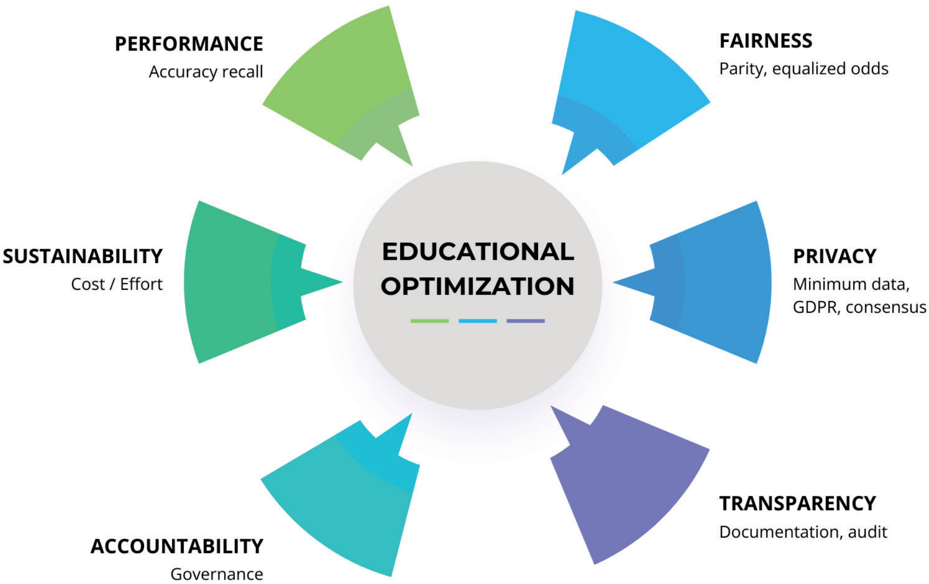


**Figure 15.1** - Conceptual compass for responsible educational optimization

Visual framework depicting how educational optimization balances six core dimensions: performance, fairness, privacy, transparency, accountability, and sustainability. The compass illustrates these as interdependent axes guiding responsible deployment of Genetic Algorithms and Evolutionary Computation within institutional decision-making.

populations of candidate settings across generations, the product of (population size) × (generations) × (evaluation cost) can dominate budgets, affecting the use of those solutions. Thus, the feasibility of EC relies on applying high-dimensional, large-scale data processing that doesn't compromise time or fidelity.

To address these issues (how big the data are and how much compute you can afford), a first good practice is to lower the cost of each evaluation before the EA even starts. Educational data is often full of redundant events and collinear signals, so shrinking the matrix pays off. Lightweight filtering and randomized linear algebra (e.g., randomized PCA/SVD) can reduce rank with subquadratic complexity and support streaming updates, making them, for example, ideal for semesterly refreshes (Halko, Martinsson & Tropp, 2011). Then a GA wrapper optimizes the remaining subset jointly with model hyperparameters. Surveys find evolutionary feature selection (Xue et al., 2016; Alias et al., 2024) effective in the high-dimensional regimes common to EDM. Other experimental results show that modified micro-GAs paired with light classifiers (e.g., decision trees) reduce feature counts and improve predictive accuracy on the evaluated datasets (Tan et al., 2017). Because simpler downstream models are faster to train, the GA can evaluate many more candidates per unit time.

In addition to reducing evaluation costs, attention should be paid to improving the efficiency of evolutionary research itself. Where non-stationarity is the rule (very commonly in education, where we have to face new cohorts, policies, and content), prequential evaluation with sliding windows is the pragmatic norm: it re-evolves only a limited subset of knobs (window length, class weights, alert thresholds), keeping compute proportional to change rather than to data size (Gama et al., 2014). Warm-starts also reduce evaluations: they allow for initializing populations from prior best individuals (e.g., those detected in the previous term) or from strong regularized-evolution baselines that encourage steady progress with small populations (Real et al., 2019). Elitist GAs (adopted in Queiroga et al., 2020 as a case) complement this approach, improving reliability while preserving top solutions across generations – useful for early-term risk detection – even if repeated high-fidelity scoring can throttle scale. To mitigate this, elitism is often paired with feature reduction and fitness caching, so that elites can be re-scored efficiently (Tan et al., 2017).

In all these situations, maintaining diversity is important to avoid premature convergence and redundant exploration. Some operators help doing this, avoiding re-exploration of the same basins while simultaneously improving sample efficiency. Examples are crowding distance in NSGA-II and speciation in NEAT-style encodings (Deb et al., 2002; Stanley & Miikkulainen, 2002).

A complementary line of defense is to exploit parallelism, the "natural friend" of evolution. Fitness evaluations are "embarrassingly" parallel: production systems commonly adopt master-worker schedulers or island models in which sub-populations evolve on separate nodes and periodically exchange elites. This shortens the necessary time while preserving diversity (Alba & Tomassini, 2002; Cantú-Paz, 2001). Modern multi-objective toolkits, such as jMetalPy (Benítez-Hidalgo et al., 2019), support parallel and distributed evaluation for MOEAs, facilitating easy distribution of evaluations to CPU/GPU clusters while keeping the algorithm logic simple. On MOOC platforms (Hao, Gan & Zhu, 2022) and all other systems that need near-real-time adjustments, this pattern is not just convenient but essential.

Further gains can be obtained in cloud-native, data-local architectures. Here, containerized evaluations, sharded datasets (by course or cohort), and asynchronous islands minimize data movement and avoid head-of-line blocking. In these cases, for example, each faculty or program can run its own island on local shards; elites migrate weekly, and only a handful of global candidates are re-evaluated centrally. Latency remains within acceptable limits for proactive outreach while keeping within institutional data boundaries. Where privacy or bandwidth limit centralization, federated patterns move code to data, exchanging model deltas or aggregate fitness summaries instead of raw records (McMahan et al., 2017).

When the "inner evaluator" is intrinsically expensive (say, sequence models over long temporal windows), a solution is to turn to surrogate-assisted evolution and multi-fidelity evaluation. A surrogate regressor (Gaussian process, random forest, or shallow net) learns to approximate the true fitness landscape, so that most offspring can be cheaply screened, and only promising candidates receive full, high-fidelity evaluation. Reviews report order-of-magnitude evaluation savings with well-designed infill criteria (Jin, 2005, 2011). Similarly, multi-fidelity schemes act like early stopping inside the EA: candidates are first judged on shorter training horizons, fewer folds, or smaller cohorts; only the top-tier solutions proceed to full training. These ideas dovetail with Population-Based Training (PBT), which mutates hyper-parameters online, prunes weak lineages, and reallocates compute to better ones converging faster under tight budgets (Jaderberg et al., 2017).

Finally, scalability must be balanced with broader objectives. Making compute an explicit objective by penalizing wall-clock time, GPU-minutes, or memory aligns evolutionary search with Green AI and ML technical debt (Schwartz et al., 2020; Sculley et al., 2015), so evolution favors solutions that meet latency and cost targets. Multi-objective frameworks such as NSGA-II and MOEA/D then surface Pareto-efficient trade-offs among accuracy, calibration, inference time, and even fairness metrics; stakeholders can choose operating points consistent with policy and infrastructure (Deb et al., 2002; Zhang & Li, 2007; Hardt, Price & Srebro, 2016; Zafar et al., 2017).

However, there is no evolutionary method that dominates all educational tasks: algorithm choice should depend on problem structure, stationarity, and budget. In such situations, hybridization is beneficial: coupling GAs with decision trees or deep networks (Tan et al., 2017; Ouyang, Zheng & Jiao, 2022) supports responsive hyper-parameter optimization.

Case patterns that can go to scale complete the engineering story depicted til now:

- in *university-scale early-warning,* teams train GRU/Transformer baselines on rolling weekly windows; a GA outer loop evolves the look-back horizon, resampling method, learning-rate schedule, class weights, and alert thresholds (Queiroga et al., 2020). Surrogates score (Jin, 2011) most offspring on a subset of courses and brief training; only the non-dominated few receive full evaluation. Islands run per faculty on a small cluster; explanation stability (e.g., SHAP consistency, Lundberg & Lee, 2017) and equalized-odds gaps are part of the fitness (Hardt, Price & Srebro, 2016).
- in *preference-aware timetabling,* GA timetablers account for student options and time zones but enforce hard feasibility. Institutions succeed by using matheuristics (i.e., metaheuristic-exact hybrids; Puchinger & Raidl, 2005): the GA proposes high-quality drafts, then an ILP (Integer Linear Programming)/CP (Constraint Programming) "repair" step satisfies hard constraints. Parallel islands explore alternative calendars per program, while surrogate feasibility checks weed out impossible offspring early (Mahlous & Mahlous, 2023).
- in *GA-enabled grouping for live courses,* instructors adjust weights for role balance, prior achievement, and students' features; a GA searches the combinatorial assignment space (Li, Ouyang & Chen, 2022; Moreno, Ovalle & Vicari, 2012). To scale, systems can precompute learner embeddings from interaction traces so

that the fitness function reduces to matrix lookups plus small scoring networks. In a few minutes, asynchronous processing generates several feasible class rosters per lab section; instructors review the rationale and can lock in any changes, resulting in better outcomes and perceived satisfaction than random or self-selected grouping.

- in *MOOC dropout prediction with micro-interventions,* platforms run hybrid GA+DL models (Zakaria, Anas & Oucamah, 2022) or GA+Bayesian Networks (Hao, Gan & Zhu, 2022) to forecast disengagement windows and optimize interventions (message type, timing, content) within rate limits. Fitness combines completion uplift from A/B tests, time-to-alert, and compute budget. PBT (Jaderberg et al., 2017) keeps the underlying nets tuned as cohorts change; islands align to course clusters.

Notwithstanding, even with mature parallelism, distributed-systems friction remains a reality: synchronization delays, heterogeneous workers, and distributed state can erode theoretical speedups. Island models and asynchronous migration can help, but robust practices also require interoperable experiment tracking (datasets, random seeds, genome definitions), reliable caching, fault-tolerant job orchestration, and clear recovery semantics. In short, scalability is not achieved through a single technique. It emerges from the combination of: (i) lean evaluators to reduce fitness costs through feature selection, surrogates, and multi-fidelity methods; (ii) disciplined search spaces and warm starts for reusing validated solutions; (iii) parallel/asynchronous execution on distributed systems with periodic synchronization; and (iv) objective functions that explicitly incorporate latency, cost, and fairness. With these pieces in place – and by reusing prior work rather than recalculating it each time – it is possible to operate EC at MOOC/VLE scale, meeting real-time needs for adaptation and scheduling while keeping accuracy and inclusivity front and center.

## 15.2 Data privacy and security concerns

Data privacy and security are central to any real-world deployment or research in education and Learning Analytics (Ungerer & Slade, 2022; Bellini et al., 2019), including, of course, cases that involve the use of Genetic Algorithms (GAs) and Evolutionary Computation (EC). Since these techniques run on rich, longitudinal data (clickstream traces, assessment histories, forum posts, demographic data, interaction logs, and more), they necessarily expose Personally Identifiable Information (PII) and sensitive educational records. This combination brings dual obligations: to obey and follow legal frameworks and laws, as GDPR (General Data Protection Regulation), and honor educational ethics protecting learners' dignity, agency, and equity (Slade & Prinsloo, 2013; Drachsler & Greller, 2016).

This section describes privacy and security not as obstacles but as design constraints to be met when establishing the GA/EC pipelines from data collection to model deployment.

## Legal and governance foundations

In the EU, the General Data Protection Regulation requires lawfulness, fairness, and transparency, purpose limitation, data minimization, accuracy, storage limitation, and integrity and confidentiality (Regulation (EU) 2016/679). Art. 25 - "Data protection by design and by default" means that systems must embed safeguards: least-privilege access, encryption in transit/at rest, retention limits. This is also true when we speak of GA/EC. See Figure 15.2 for an application in this case for privacy preservation before optimization begins. Two practice-oriented frameworks help operationalize it: (i) the already cited DELICATE checklist in Learning Analytics (Determine, Explain, Legitimate, Involve, Consent, Anonymize, Technical, External) for transparency and stakeholder engagement (Drachsler & Greller, 2016), and (ii) institutional information-security controls, such as ISO/IEC 27001 and NIST SP 800-207 (Zero Trust), align AI-risk governance with NIST AI RMF 1.0 (ISO/IEC, 2022; NIST, 2020, 2023).

## Anonymization, pseudonymization, and re-identification risk

A first instinct is to "anonymize the data and proceed". Putting this into practice, de-identification is not straightforward. Just replacing direct identifiers with tokens in a data set, the so-called pseudonymization, still involves some risk of linkage where quasi-identifiers exist. Re-identification attacks on "anonymized" datasets (e.g., movie ratings, Narayanan & Shmatikov, 2008) show how sparse educational traces can be. Robust anonymization requires techniques such as k-anonymity, l-diversity, or t-closeness, each with trade-offs for utility and privacy (Article 29 Data Protection Working Party, 2014; Sweeney, 2002). In light of all these observations, anonymiza-

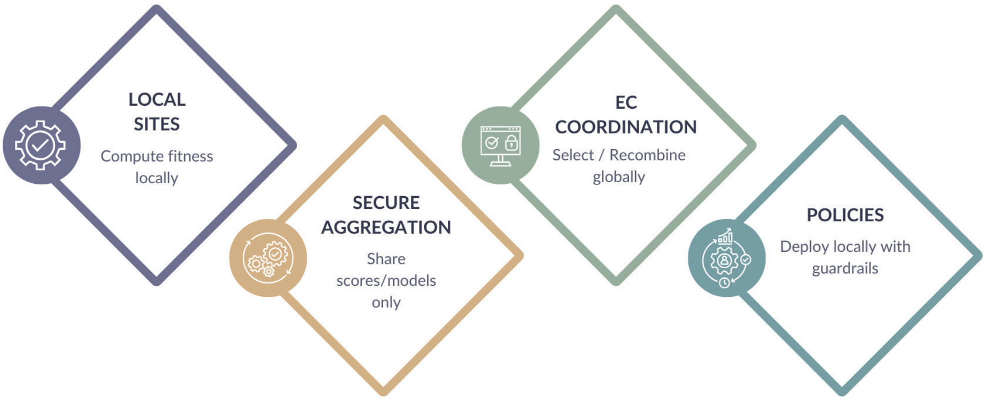


**Figure 15.2** - Privacy preserving / Federated Evolutionary Optimization

Federated EC: local fitness evaluation with secure aggregation and centralized evolutionary coordination to protect privacy.

tion should not be used in isolation but should be accompanied by technical privacy guarantees and process controls.

### Differential Privacy (DP) and privacy-preserving training

DP adds calibrated noise so that results are statistically indistinguishable with or without any one learner's data; it composes well across repeated queries and model updates (Dwork & Roth, 2014). DP-SGD (Differentially Private Stochastic Gradient Descent algorithm) has been proposed for deep learning, enabling the bounds on privacy loss (the ε "budget") in neural architectures (Abadi et al., 2016). Building on this technical notion of a cumulative privacy budget, applying evolutionary model selection procedures, institutions could set ε targets by policy (e.g., lower for high-stakes risk alerts), log cumulative privacy loss, and integrate the ε budget as a constraint inside the GA fitness, so that a solution exceeding budget is penalized, whilst accuracy calibrated under DP noise is rewarded.

### Federated Learning (FL) and secure aggregation

Where regulations or culture discourage central data collections on students, FL is a method that creates models locally and sends updates to central servers; raw records do not leave the device or campus. To counter the risk that model updates themselves leak information, secure aggregation protocols hide each client's update from the server (Bonawitz et al., 2017). The GA/EC "outer loop" can be moved to the server (for hyperparameter and architecture selection), and gradient-based training remains on clients, aligning with GDPR's data minimization and purpose limitation principles (McMahan et al., 2017; Kairouz et al., 2021). These solutions also contribute to balancing privacy and scalability: for at-scale platforms (MOOCs, virtual universities), federated or sharded evolution (islands per program/course) reduces cross-domain exposure and aligns with institutional data boundaries, while secure aggregation curbs leakage during update collection.

### Threat models specific to GA/EC pipelines

Even with strong management, considering intermediate models and continuous evaluations on which they are built, GA/EC introduces attack surfaces beyond standard analytics:

- membership inference can reveal whether a student's data were used to train a model, especially for overfitted networks; mitigation includes DP, regularization, and calibration (Shokri et al., 2017; Abadi et al., 2016).
- model inversion can use outputs or confidence scores to reconstruct some features about individuals: mitigation strategies involve limiting confidence disclosures and applying DP (Fredrikson, Jha & Ristenpart, 2015).
- leaks in surrogate-assisted evolution arise when the approximation model stores sensitive signals; mitigation includes training surrogates on DP-perturbed summaries and purging caches after the search phase.

- pipeline logs (hyperparameters, fitness traces, intermediate artifacts) can inadvertently store PII if not scrubbed, especially when GA fitness functions directly use behavioral features (e.g., forum keywords); mitigation methods require enforcing redaction and retention limits at the logging layer.

## Consent, notice, and learner agency

Ethically defensible optimization requires informed consent and meaningful notice. Learners should know what data is collected, for what purposes, for how long, and with which implications (Slade & Prinsloo, 2013; Drachsler & Greller, 2016; Bellini et al., 2019). Practical steps include tiered consent and opt-out pathways that preserve access to core services. To explain: in notices, we need to introduce separate checkboxes for research vs. operational analytics and give learners the opportunity to use some (or, better, all) services even if they decline data collection. Where GA-optimized decision rules trigger interventions, institutions should provide "meaningful information about the logic involved" and allow contestation (Regulation (EU) 2016/679, GDPR, Arts. 13-15). As an example, in content sequencing (Christudas, Kirubakaran & Thangaiah, 2018), simple, plain-language rationales build trust: "we recommended Topic X because you mastered Y and struggled with Z". Additionally, accountability practices can support learners in understanding how decisions are generated and what constraints shape them.

## Bias and fairness in optimization objectives

Privacy protections alone do not mean that fairness prevails. Training data often under-represent subgroups, and naively optimized objectives can widen gaps (Ungerer & Slade, 2022). Two practices help: (i) auditing datasets and outputs using disparity metrics such as false-negative gaps or equalized odds, and (ii) embedding fairness directly into the multi-objective fitness of GA. NSGA-II or MOEA/D can surface Pareto-efficient trade-offs among accuracy, calibration, intervention workload, and fairness gaps, enabling policy-guided choices (Hardt, Price & Srebro, 2016; Zafar et al., 2017; Deb et al., 2002; Zhang & Li, 2007). Group-aware reweighting and other data augmentation strategies, and targeted sampling can further reduce bias, but should be documented and reviewed by cross-functional ethics committees.

## Data minimization and purpose limitation in practice

GA/EC practitioners should start with only the features necessary for the stated purpose and any such expansion needs to be justified through a Data Protection Impact Assessment (DPIA) (Regulation (EU) 2016/679, GDPR, Art. 35). In pipelines that use event vocabularies like xAPI/Caliper, some indications can be strip or hash free-text fields early; store only derived aggregates for modeling; and set explicit retention horizons, e.g., purge raw logs after *n* days, retain model-ready summaries for *m* terms (Advance Distributed Learning, 2016; 1EdTech, 2020). Where text features are indispensable (e.g., help-request semantics), consider on-device vectorization with DP noise before upload.

### Accountability, Transparency, and Reproducibility

Accountability mechanisms ensure transparency, reproducibility, and auditability in data-driven, evolutionary systems and enable learners and practitioners to understand workflows and decisions in data usage. Examples include (i) documenting origins, characteristics, and limitations of data and models through dataset datasheets (Gebru et al., 2021) and model cards (Mitchell et al., 2019); (ii) versioning genome representations and fitness definitions to track how solutions evolve and associate configurations with specific outcomes; (iii) recording random seeds to support reproducibility; and (iv) setting explicit budget caps for compute time, or costs to adapt resources to operational and ethical requirements.

### Putting it together: an operational template

Before launch, conduct a DPIA; define the minimal feature set; select a privacy technology stack (DP targets, FL topology, secure aggregation); and codify fairness metrics within GA fitness. During operation, monitor privacy budgets (ε) and fairness gaps; rotate secrets and cull caches after searches. Periodically (e.g., after each term), publish a brief transparency report: what changed in the genome, how many alerts were generated, how fairness/accuracy moved, and how privacy budgets were spent. This cycle aligns technical safeguards with the ethical imperative to treat learners not as mere data subjects but as co-owners of their educational trajectories.

## 15.2.1 Security engineering for educational cloud environments

Massive open online course (MOOC) platforms, as well as institution-wide virtual learning environments (VLEs), aggregate huge amounts of assessment traces, click-stream logs, and communications: they are attractive, high-value targets, but their conformation also raises operational risk. Thus, security architecture should adopt a defense-in-depth approach: layered protections (network segmentation, intrusion detection and prevention systems, hardening of managed services) combined with a zero-trust model where every request is authenticated and authorized, assuming the network is inherently hostile (NIST, 2020). Concretely, this includes encrypting data in transit (TLS 1.3) and at rest, using hardware security modules for key management, isolating GA workers in containers with strict CPU/GPU quotas, and enforcing role-based access control with just-in-time privileges. GA/EC workloads are distributed across multiple workers; secrets such as dataset credentials and keys must be securely vaulted, regularly rotated, and never embedded in genome definitions or experiment manifests. Immutable audit logs, routine patch management, and regular penetration testing and third-party audits are essential, particularly when xAPI or Caliper feeds expose detailed interaction events (Advanced Distributed Learning Initiative, 2016; 1EdTech, 2020).

In cases where GA/EC capabilities are provided by third-party processors (such as grouping plug-ins or adaptive sequencing tools), contracts should specify security requirements, incident-reporting SLAs (Service Level Agreements), and tacit obligations to return or erase data upon termination of service. These practices align with GDPR's security and accountability principles, as well as with the NIST AI Risk Management Framework's emphasis on identifying, assessing, and mitigating AI-related risks.

### 15.2.2 LLMs and generative AI in GA/EC pipelines: distinct privacy risks

We have already described in this book that evolutionary search can be paired with Large Language Models (LLMs), for example, by using an LLM-based tutor whose prompts or fine-tuning parameters are evolved by a GA.

The EU AI Act (Regulation (EU), 2024/1689) establishes a risk-based regime for AI. Systems used to determine access to education, to evaluate students, or to guide their progression are designated high-risk, triggering stringent requirements (e.g., risk management, data governance, technical documentation, human oversight, robustness, and logging). For general-purpose AI (GPAI)/foundation models (including LLMs), the Act adds transparency obligations (e.g., technical documentation, copyright-related disclosures) and heightened duties where a model presents systemic risk (European Union, 2024; EDPB, 2024).

Here, LLM-specific privacy and data-protection challenges (NIST, 2023; EDPB, 2024) follow:

- *training data provenance and legal basis.* The EDPB's ChatGPT Taskforce report highlights that controllers must identify a valid GDPR legal basis for large-scale training and emphasizes transparency regarding data sources and reasonable steps in order to ensure the accuracy of personal data when outputs concern individuals (EDPB, 2024). In terms of education, this means documenting whether any student data ever enters pre-training, fine-tuning, or RAG indices, and clarifying the lawful basis, retention limits, and data-subject rights for access, objection, or erasure where applicable.
- *output privacy and model leakage.* Models may memorize or reconstruct sensitive snippets even when the training data excludes PII. Institutions should prohibit the inclusion of any student-identifying data in LLMs hosted by third-party providers, except when a data protection impact assessment (DPIA) has been conducted, and appropriate processing terms are put in place; furthermore, they should implement PII redaction and prompt protection when student work is analyzed.
- *accuracy and explainability.* Because LLMs can hallucinate, any use in learner assessment or risk triage must include human-in-the-loop review and accuracy guarantees that scale up to the decision's impact, as mandated by the GDPR's fairness and accuracy principles.

### 15.2.3 Operational implications

If a GA-optimised early-warning model informs significant decisions (e.g., mandatory interventions), it should be clearly defined as a high-risk use case requiring a risk management system, checks on quality and representativeness of the training data, event logging, and provisions for human oversight. If your workflow embeds an LLM (e.g., to generate tailored study plans) and the LLM is a GPAI (General Purpose AI) model, you need to ensure that the provider is compliant with GPAI transparency duties and that any downstream fine-tuning or integration maintains compliance. In both cases, you should align GDPR DPIAs with AI Act risk files so the same hazards (privacy, bias, robustness) are tracked and mitigated consistently across regimes.

A practical privacy & security checklist for GA/EC in education:

- *purpose & legal basis:* define specific purposes; pick and document a lawful basis; complete a DPIA before pilot.
- *minimisation & retention:* collect the minimum data necessary; set short, documented retention for raw logs and GA caches.
- *access control & encryption:* encrypt at rest/in transit; enforce least-privilege access; audit all data/model access.
- *privacy-preserving design:* prefer federated/island evaluations; consider differential privacy on aggregate metrics; test de-identification robustness.
- *fairness & transparency*: include fairness and calibration in GA fitness; publish model cards; enable effective human review paths.
- *AI Act mapping*: determine whether the use is high-risk; if GPAI/LLM is involved, verify provider obligations and keep technical documentation and logs as required.

In sum, protecting learners' privacy while deploying GA/EC at scale is entirely feasible but it demands security-by-design, privacy-preserving engineering, rigorous governance, and now, AI-Act-aware compliance. Doing this work up front not only reduces legal exposure; it builds trust and improves model quality by forcing clarity about purpose, data provenance, and acceptable trade-offs.

## 15.3 Ethical implications of algorithmic decision-making

Algorithmic decision-making in educational systems driven by Genetic Algorithms (GAs) and Evolutionary Computation (EC) raises complex, multifaceted ethical considerations related to equity, inclusiveness, transparency, and accountability. These systems offer powerful optimization capabilities, yet they also pose substantial ethical risks that must be anticipated and mitigated if they are to be deployed fairly in real institutions.

In earlier paragraphs, we have introduced fundamental documents in the European context, such as the General Data Protection Regulation (GDPR). It includes provisions that restrict decisions from being based solely on automated processing and grants data subjects the right to meaningful information about the logic involved. Likewise, the EU Artificial Intelligence Act (AI Act) further codifies lifecycle obligations for high- and limited-risk educational AI systems-risk management, data governance, human oversight, and transparency, setting a higher bar for responsible deployment across the sector (Regulation (EU) 2024/1689). The principles from the two regulations have to be applied directly to GA-driven tools and hybrid solutions such as early-warning or automated placement systems (Regulation (EU) 2016/679).

The bias in the datasets on which GA/EC models are evolved or tuned is a consistent issue, and that, when unaddressed, can have impacts that can perpetuate inequity with unfair outcomes: under-representation of certain cohorts (e.g., first-generation students, older learners, learners with disabilities) can skew learned rules or fitness functions, causing personalized pathways or risk predictions to underserve those very groups. Methodologically, the optimization goals of GA/EC systems can incorporate several families of fairness criteria, for instance, equality of opportunity or equalized odds, which bound error-rate disparities across protected groups (Hardt, Price & Srebro, 2016), and constraints against "disparate mistreatment", which limit group-conditioned error differences during training (Zafar et al., 2017). This means that, in practice, including fairness gaps and calibration error as explicit terms of the multi-objective fitness function steers a search for Pareto-efficient solutions more effectively toward the values of educational efficacy and justice. These technical adjustments also align with the EU AI Act's requirement to manage fundamental rights risks throughout the AI lifecycle (Regulation (EU) 2024/1689).

Transparency is another major ethical concern. Many GA/EC pipelines wrap complex inner learners (ensembles or deep nets), making it hard for educators and students to understand why a placement, grouping, alert, or resource decision was made. Explainable AI (XAI) tools, such as SHAP which provides locally faithful attributions for individual predictions (Lundberg & Lee, 2017), can be layered onto GA-optimized models to produce case-by-case rationales that students and advisors can interrogate. Complementary documentation practices help make systems auditable and governable at the institutional level. A kind is represented by model cards that explain what the model does, how it was trained on data, documented biases, and subgroup performance (Mitchell et al., 2019). In EU jurisdictions, these practices align with GDPR expectations for meaningful information about automated logic and with the AI Act's transparency and human-oversight duties. Robust human-in-the-loop checks and clear explanations are essential to avoid automation misuse or over-reliance (Parasuraman & Riley, 1997; Bahner, Hüper & Manzey, 2008). Counterfactual explanations (Wachter, Mittelstadt & Russell, 2017) and actionable recourse methods (Ustun, Spangher & Liu, 2019) help students understand and contest decisions.

Accountability is an additional ethical pillar, complicated by the iterative, self-optimizing nature of EC. When a GA-tuned model incorrectly flags a student as low-potential and/or misallocates scarce tutoring hours, who is responsible for the consequences: the developer, the data steward, the teachers, or the institution? Good practice distributes accountability across these roles and requires traceability: this means version-controlled search spaces, datasets, seeds, and fitness definitions, plus audit logs of human overrides and rationale. GDPR complements this with clear controller-processor responsibilities and rights of access, explanation, and contestation for affected learners. Sector regulators underscore that solely automated decisions in high-stakes contexts should be avoided or paired with strong human review (Information Commissioner's Office, n.d.; Regulation (EU) 2016/679). These same expectations are mirrored in the AI Act's provisions on post-market monitoring and incident reporting for high-risk systems (Regulation (EU) 2024/1689).

Let's apply these principles to some practical uses in group formation and resource allocation.

GA-based grouping has been shown to improve task performance by optimizing complementary skills and roles (Moreno, Ovalle & Vicari, 2012; Li, Ouyang & Chen, 2022). However, the use of GAs to form collaborative learning groups exposes a related ethical risk: the possibility of perpetuating stereotypes or neglecting valuable but less easily quantifiable characteristics, like creativity and resilience. Fitness functions must be constructed carefully to avoid over-privileging a narrow band of attributes. Besides, grouping criteria have to be broadened (including, for instance, prior achievement together with participation histories and role diversity) to improve learning outcomes and perceived fairness, suggesting that inclusive fitness definitions can systematically counter stereotyping effects. Specifically, this means including diversity-sensitive metrics (not only immediate performance gains) in the GA's fitness function and performing an impact audit of group assignments to avoid disparate harm before deploying at scale.

Efficiency-inclusivity trade-offs are unavoidable in GA-driven resource allocation and curriculum optimization. Multi-objective formulations that are not carefully constrained can easily shift to cost-minimizing or throughput-maximizing solutions that inadvertently fail students from lower-resource backgrounds. So, the possible misaligned use of GA/EC in such contexts requires particular precautions: optimizing only for institutional outcomes (e.g., completion rates or cost per credit) risks increasing inequities if the models provide resources to "likely completers" while shortchanging those with stronger needs, an instance of disparate mistreatment. Fairness-aware optimization (Hardt, Price & Srebro, 2016; Zafar et al., 2017) addresses these risks through explicit equity terms, such as penalizing disparate false-negative rates in early-warning systems or prioritizing accessibility objectives (e.g., disability accommodations or time-zone fairness in online timetables) alongside traditional performance metrics (e.g., ensuring that tutoring resources are not disproportionately assigned to already high-performing students, or that intervention thresholds do not systematically exclude part-time or working learners). Beyond design-time constraints, ongoing monitoring is essential. Periodic impact assessments should evaluate subgroup outcomes, not only aggregate performance, and establish triggers for retraining or recalibration when inequities emerge (e.g., detecting if certain demographic groups consistently receive fewer support interventions despite similar risk levels, or if curriculum sequencing disadvantages students in specific time zones or with accessibility needs). Embedding fairness constraints within both the search process and post-hoc evaluation pipelines ensures that optimization remains sensitive to distributional effects over time. Empirically and normatively, this approach supports solutions that remain efficient while no longer being blind to equity considerations, aligning algorithmic decision-making with institutional missions and with legal obligations under GDPR and the EU AI Act (Regulation (EU) 2016/679; Regulation (EU) 2024/1689).

In summary, GA/EC can help institutions personalize learning, triage support, and schedule resources more intelligently but only if ethical guardrails are integral to design, not an afterthought.

Concretely, this means (i) data audits and representativeness checks; (ii) fairness-aware fitness functions and reporting; (iii) explanation tooling and model documentation; (iv) human-in-the-loop governance and override pathways; and (v) rigorous accountability with audit trails, all within the legal scaffolding provided by GDPR and the EU AI Act (Information Commissioner's Office, n.d.; Mitchell et al., 2019; Regulation (EU) 2016/679; Regulation (EU) 2024/1689).

We closed this chapter by providing a card template (Table 15.1) for governance listing basic solutions to adopt for a fair and auditable education system when using GA.

**Table 15.1** - Optimization Card template for governance

Documentation fields (objectives, constraints, fairness metrics, and monitoring) for transparent audit of GA/EC educational systems.

| Field | Description |
|---|---|
| Objectives & weights | List of objectives, weights (or MOEA objectives) |
| Constraints | Hard/soft constraints and how enforced |
| Data & evaluation | CV protocol, test periods, lead time |
| Fairness metrics | Definitions, subgroup sets, thresholds |
| Selected solution | Why chosen (knee point / committee decision) |
| Monitoring | Drift, recalibration, incident handl |

# Chapter 16

## Future opportunities and emerging trends

As educational landscapes evolve, the integration of advanced computational techniques into research and practice, especially here Genetic Algorithms (GAs) and the broader family of Evolutionary Computation (EC), opens substantial opportunities for innovation (Ifenthaler & Yau, 2020; Ouyang, Zheng & Jiao, 2022; Papamitsiou & Economides, 2014; Charitopoulos, Rangoussi & Koulouriotis, 2020; Kučak, Juričić & Đambić, 2019; De Santis & Minerva, 2026).

The sections that follow examine and summarize how these methods are interwoven with Artificial Intelligence (AI) and big data infrastructures, and how they enable formal education as well as unexplored forms of lifelong learning and workforce development. We emphasize three themes throughout: (i) scalable personalization (adapting content, supports, and pacing to each learner at cohort scale through federated learning and master-worker/island-model evolution), (ii) operational feasibility (running real systems within bounded latency and budget), and (iii) equity-by-design (embedding fairness, accessibility, and inclusion goals directly into optimization objectives). In doing so, we highlight how GA/EC methods can shape educational practice to better accommodate diverse learners in a rapidly changing world.

### 16.1 Integration of GAs with AI and big data

Taken together, hybrid GA-AI designs, distributed computation, lean evaluators, and standards-driven telemetry allow routinely scalable personalization in online and blended programs. On the governance side, there is an obligation associated with the technical upside: designing fairness, privacy, and transparency and reduce compute costs through fitness functions and optimization processes (Hardt, Price & Srebro, 2016; Zafar et al., 2017; Slade & Prinsloo, 2013). We'll now examine how genetic algorithms, and especially hybrid models, leverage big data and address scalability in educational contexts.

## Hybrid GA-ML pipelines for prediction and grouping

Merging GAs with machine learning models can improve both predictive accuracy and operational efficiency in Learning Analytics (LA) and Educational Data Mining (EDM). Hybrid designs apply GAs as wrappers to select compact, informative feature subsets and to tune hyperparameters of a downstream learner (e.g., tree ensembles, gradient boosting, or neural networks). In a large institutional context (46 courses over 24 consecutive weeks), Tan, Lim, Bong, and Liew (2017) report that a modified micro-GA combined with decision trees reduced the number of input features, leading to an improvement in classification accuracy for at-risk prediction. This “shrink-then-tune” pattern reduces computational overhead, enhances generalization, and aids later monitoring. Applying the two methods in reverse order, Shin-ike and Iima (2008) predicted students’ learning outcomes with a neural network model, then applied a Genetic Algorithm to the resulting data to identify which pairs would collaborate best.

## Big-data scale through parallel and distributed evolution

Large platforms (e.g., MOOC providers, institution-wide VLEs) generate high-volume, high-velocity data-clickstreams, assessments, discussion fora, proctoring traces, often across tens of thousands of learners per term (Mir & Khan, 2026; Stracke et al., 2019; Reich & Ruipérez-Valiente, 2019; Ruipérez-Valiente et al., 2022; Reich, 2022). Because fitness evaluation in EC is “embarrassingly” parallel, GA/EC pipelines can scale via master-worker and island models that evaluate many candidate solutions concurrently, exchanging elites periodically to preserve diversity (Alba & Tomassini, 2002; Cantú-Paz, 2001). Asynchronous, distributed execution allows institutions to farm out evaluations to CPU/GPU clusters without rewriting core algorithm logic (Benítez-Hidalgo et al., 2019). In retention modeling for MOOCs (Hao, Gan & Zhu, 2022; Zakaria, Anas & Oucamah, 2022), for example, distributed evolution enables near-real-time recalibration of hyperparameters (window length, class weights, alert thresholds) while keeping inference latency within service-level targets.

## Lean evaluators: feature selection, surrogates, multi-fidelity

At big-data scale, cost per evaluation is as important a consideration as parallelism. Two-stage pipelines, composed of fast filters and randomized PCA/SVD to condense the space and reduce dimensionality (Halko, Martinsson & Tropp, 2011), followed by a GA wrapper that jointly optimizes the remaining features and model hyperparameters, are particularly well-suited to parallel execution and common in education, where signals are redundant and collinear. Evolutionary feature selection (Xue et al., 2016; Alias et al., 2024) is competitive in the high-dimensional regimes typical of EDM. When inner learners are expensive (e.g., long-horizon sequence models), surrogate-assisted evolution (Jin, 2005, 2011) and multi-fidelity evaluation (shorter training, subsampled courses/cohorts) curb costs without sacrificing final quality: only the most promising offspring graduate to full-fidelity training. Population-Based

Training (PBT) further reduces evaluation cost by accelerating convergence through online hyperparameter mutation and reallocating compute to better lineages (Jaderberg et al., 2017).

### From GA-tuned predictors to adaptive, human-centered systems

Beyond prediction, GAs act as controllers of adaptive learning systems. Interactive evolutionary systems use visual analytics (Llorà et al., 2006) to expose what the fitness function values (e.g., balancing cognitive load and engagement), enabling “glass-box” adjustments by instructors. In intelligent tutoring (Woolf, 2009; VanLehn, 2011) and formative assessment, GA/EC can adapt sequencing and schedules, with human-in-the-loop overrides to maintain pedagogical coherence. These designs align with evidence that transparent personalization fosters trust and uptake, an important lesson for production LA/EDM (Ifenthaler & Yau, 2020).

### Curriculum and timetable optimization at platform scale

Scheduling and curriculum optimization are classic EC successes. At platform scale, these systems operate over large, heterogeneous datasets, such as enrollments, room and teacher availability, making timetabling a natural intersection of evolutionary optimization, AI-driven decision support, and big data analytics. Recent surveys of educational timetabling consolidate matheuristic and EC advances, cataloging state-of-the-art results across constraints and benchmarks (Ceschia, Di Gaspero & Schaerf, 2023; Dunke & Nickel, 2023; Bashab et al., 2023). Preference-aware GA timetabling could incorporate student choice and time-zone fairness and then hand solutions to exact solvers (ILP/CP) for final repairs (Mahlous & Mahlous, 2023; Puchinger & Raidl, 2005). Incorporating equity metrics into the GA’s fitness function (for example, promoting accessibility or minimizing operational conflict for carers or working students) or into a multi-objective evolutionary algorithm embeds operational calendars into inclusion goals.

### Evolving neural networks and end-to-end pipelines

GAs are well-suited to global exploration of deep model design spaces – layer counts, connectivity, learning rates, or augmentation policies – reducing the brittleness of manual tuning (Goldberg, 1989; Stanley & Miikkulainen, 2002; Harp, Samad & Guha, 1989). Regularized evolution and related methods have found competitive architectures under bounded budgets (Real et al., 2019). In educational data, this yields robust predictors of engagement and success that remain calibrated under cohort drift while preserving the option to multi-objectivize the search for both accuracy and latency/fairness (Deb et al., 2002; Zhang & Li, 2007).

### LLMs + Evolution: a new frontier

Recent work views language models themselves as optimizers or components inside evolutionary loops. “LLMs as optimizers” (OPRO) prompts models to iteratively improve candidate solutions, sometimes rivaling gradient-based baselines in constrained settings (Yang et al., 2024). EvoPrompt connects EC with LLMs to search over prompts, tool-use plans, and controller policies (Guo et al., 2024), whereas other systems, such as PromptBreeder, evolve prompts self-referentially to improve task performance over time (Fernando et al., 2023). For education, these ideas point to layered systems where (i) an LLM tutors or generates feedback; (ii) a GA optimizes the LLM's policy (prompting, content mix, timing) for specific courses and cohorts; and (iii) multi-objective fitness balances learning gains, student workload, fairness, and compute. Of course, such pipelines must still adhere to sector governance and privacy norms (Drachsler & Greller, 2016; Slade & Prinsloo, 2013; Ungerer & Slade, 2022; European Union, 2016, 2024; EDPB, 2024), but they create an exciting runway for adaptive, teacher-augmenting tools at scale.

### Standards, data governance, and reproducibility

Big-data GA/EC depends on clean, portable telemetry. Open specifications like xAPI and 1EdTech Caliper make it possible to stream learning events into analytics safely and consistently (Advance Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; 1EdTech Consortium, 2020). Reproducibility and thoughtful review on the modeling side are enabled by disciplined experiment tracking, like versioned genomes, fixed seeds, dataset snapshots, and model cards (Mitchell et al., 2019). As EC increasingly cohabits with LLMs and deep models, these scaffolds will become a must-have.

## 16.2 Applications in lifelong learning and workforce training

Up to now, we have focused mainly on the applications of GAs and EC in higher education. Across higher education and work, the application is consistent: GA/EC can improve targeting, personalization, and efficiency; evolutionary feature selection can yield leaner, more generalizable models; and multi-objective optimization can align learning, cost, and equity goals. Below, we will explore how GA solutions might be adapted for corporate training on at-scale platforms and for lifelong learning.

### Skill-gap diagnostics and targeted interventions

EDM models developed in higher education for predicting at-risk students can be translated to workforce contexts with minor adjustments. GA-driven solutions include identifying the most relevant predictors and reducing dimensionality (Sahlaoui et al., 2024) or setting model hyperparameters (Queiroga et al., 2020). In this domain, GAs improve feature selection (what data matters) and threshold calibration (who needs support now), yielding precise identification of emergent gaps in work teams. Identification is the first step toward automated or human-guided interventions to get the student back on track. Video learning analytics (Hasan et al., 2020) add a complementary lens: by mining pause/replay patterns and error checkpoints, GA-tuned models can highlight hard segments and propose micro-remediations.

### Personalized upskilling at scale

Using GA for learning personalization allows aligning training modules with individual goals and organizational needs. By optimizing which concepts to prioritize and how to sequence them, EC can design learning pathways that maximize performance gains within resources and time budgets. In practice, a GA controller can pick the next activity for students based on their recent performance, pace, and preferences, updating the plan as new evidence arrives. This approach, natural in higher education, supports the acquisition and continuous updating of skills through structured learning pathways and can be easily adapted to corporate training, where such adaptive pipelines can be automatically deployed across large cohorts of learners/employees (Hssina & Erritali, 2019; El Yessefi et al., 2024).

### Predictive maintenance for skills

Skills decay without practice. Evolutionary controllers can schedule refreshers like a spaced-repetition system tuned for workplace tasks, optimizing (and thereby customizing) training intervals and modalities for each employee by balancing retention, workload, and cost. Whenever tasks change, the controller rebalances the plan, ensuring investments track actual demand.

### Adaptive, hybrid modality and accessibility

GAs can support the orchestration of hybrid training (Christudas, Kirubakaran & Thangaiah, 2018; Bhattacharjee et al., 2018), deciding which modules are self-paced, live virtual, or in-person-based on job role, connectivity, language, and accessibility needs. This "modality allocation" can be optimized to give predictive signals that inform allocation decisions (He, 2022; Qiu, 2022). The same controllers can schedule mentorship and coaching resources, ensuring those with the largest gaps get timely, high-bandwidth support.

### Multi-objective optimization under real constraints

Workforce programs contend with competing priorities: cost, throughput, satisfaction, compliance, and equity. Multi-objective GA frameworks such as NSGA-II or MOEA/D surface a Pareto front of training plans, enabling leaders to pick an operating point that balances these objectives, rather than maximizing a single metric (Deb et al., 2002; Zhang & Li, 2007). Empirically, this solution stabilizes policies across cohorts by reducing their dependence on a single outcome measure. So, a plan that yields slightly lower test gains but reduces time-to-competency and improves fairness across groups, beyond being effective, may be preferable at scale.

### From static dashboards and analytics to closed-loop improvement

Using analytics and connecting them to action can lead to the biggest gains. EC can turn static dashboards and post hoc descriptive analytics into controllers that continuously test and adapt interventions. For instance, a GA can A/B test nudge content and timing, mutating message variants, and allocating traffic based on uplift, while penalizing advisor overload. This is a concrete example of optimization with human constraints in the fitness that has to be applied in real time, not retrospectively, to be effective in improving teaching practices. These loops enable continuous improvement, not just periodical redesigns.

### Scalability and governance in enterprise settings

Large enterprises face similar scale and privacy challenges as universities. This makes the solutions already used in university settings applicable and effective in business contexts as well. These include distributed evolution (island models per business unit), federated learning patterns (moving code, not data), and fairness-aware objectives to ensure that optimization is compliant with both policy and equity constraints (McMahan et al., 2017; Hardt, Price & Srebro, 2016). Similarly, documentation practices – model cards (Mitchell et al., 2019) and audit logs – can support compliance and build trust with employees.

## 16.3 Promoting inclusivity and equitable education

Traditional models can miss signals in socioeconomic status, caregiving load, disability accommodations, or language background that can affect engagement and outcomes in training activities. Evolutionary wrappers help identify the most relevant combinations for specific targets, improving the sensitivity of early-warning and success models for underrepresented learners. In the approach of equity-by-design, we need to bake fairness into the objective through equality of opportunity (equal true-positive rates across protected groups) and equalized odds (balancing false-positive/negative rates) as explicit constraints or co-objectives (in MOEA) during GA search (Hardt, Price & Srebro, 2016; Zafar et al., 2017). This operates a shift in the Pareto front towards solutions that improve accuracy and reduce disparities.

### Inclusive resource allocation and timetabling

Otherwise, in this book, we discussed that EC tools extend beyond prediction to the allocation of constrained resources, advising time, tutoring seats, labs, lectures, or exam slots (Ceschia, Di Gaspero & Schaerf, 2023; Mahlous & Mahlous, 2023; Ngo et al., 2021). Preference- and accessibility-aware fitness functions can help institutions direct scarce capacity to those with the highest barriers. In online settings, the same principles can be applied to govern system resources (e.g., bandwidth windows for proctored exams) and ensure that students with unstable connectivity are not disadvantaged, avoiding discrimination. Optimizing such logistics has measurable impacts on inclusion and sense of belonging.

### Group formation that broadens participation

Another topic addressed several times concerns group creation: if they are appropriately composed, collaborative learning becomes quite a powerful equalizer. GA-based grouping that optimizes not only prior achievement and roles but also diversity of perspectives and equitable participation can raise artifact quality and perceived fairness (Li, Ouyang & Chen, 2022; Moreno, Ovalle & Vicari, 2012; Ugarte et al., 2022). Practically, this involves (i) adding diversity-aware terms to the fitness, (ii) exposing rationales to instructors and students, and (iii) instrumenting metrics of participation for post-hoc audit. These strategies mitigate the reinforcement of stereotypes and enhance outcomes for learners who have experienced systemic disadvantages and marginalization.

### Personalization for accessibility

GA/EC can be designed in line with UDL - Universal Design for Learning (CAST, 2018), which encourages multiple means of engagement, representation, and action. For example, they can select accessible content based on variants such as captions, transcripts, and alternative formats (Christudas, Kirubakaran & Thangaiah, 2018). Besides, they can work on sequencing and recommendations (Hssina & Erritali, 2019; Bhattacharjee et al., 2017) by adapting pacing to executive function and scheduling breaks to optimize both mastery and cognitive load. This might increase completion and satisfaction for neurodiverse and disabled students without harming overall throughput.

### Transparency, privacy, and legitimacy

Inclusion also depends on trust. Students and instructors need to understand why a decision was made and have avenues to challenge it. Techniques like SHAP for local explanations, combined with model cards, help clarify model behavior (Lundberg & Lee, 2017; Mitchell et al., 2019). Privacy-preserving patterns – federated training, secure aggregation, and strict data minimization (McMahan et al., 2017; Kairouz et al., 2021; Regulation (EU) 2016/679) – are essential to protect sensitive attributes used for fairness checks (Drachsler & Greller, 2016; Slade & Prinsloo, 2013). Finally, periodic subgroup audits make disparities visible and actionable (Hardt, Price & Srebro, 2016).

### Documented gains and remaining gaps

Across case studies, GA/EC has improved resource management and participation balance and increased satisfaction when objectives reflect fairness. Work remains to standardize subgroup reporting, evaluate long-term equity impacts on inclusion, and ensure that optimization never trades away access for efficiency.

## 16.4 Human-in-the-loop design and teacher augmentation

Even as Genetic Algorithms (GAs) and broader Evolutionary Computation (EC) scale across learning analytics pipelines, the most durable gains in practice come from framing these systems as tools that support teachers' work, rather than replace it. Through human-in-the-loop (HITL) design, educators' judgment is treated as a first-class input to optimization, acknowledging that much of what matters in education (e.g., formative intent, socio-emotional context, local policy) and in educational research (e.g., observation, replication, measurement) is tacit, situated, and not fully captured in data (De Santis, 2022; Ary et al., 2010). Human oversight is also a regulatory requirement for high-risk educational AI in many jurisdictions: the EU AI Act mandates appropriate human oversight for high-risk systems (e.g., learner profiling, automated assessment), including the ability to intervene, override, and contest outcomes (European Union, 2024). Similar lessons emerge from decades of research on intelligent tutoring (VanLehn, 2011; Woolf, 2009): the most effective systems create partnerships between instructor and automation, combining algorithmic precision with pedagogical intent.

### Decision support, not decision replacement

The human decision-making role in EC optimization processes aligns with the HITL paradigm. Evolutionary search proposes diverse candidate solutions: instructors steer the search by (a) attributing weights in the objectives (e.g., "minimize false negatives on at-risk students", "reduce advisor workload", "protect time-on-task for labs"), (b) constraining feasible regions (e.g., "do not schedule past 18:00"), and (c) flagging unacceptable trade-offs for exclusion on ethical grounds (e.g., disparate error rates across demographic groups). In multi-objective evolutionary algorithms (Deb et al., 2002; Zhang & Li, 2007), which remain excellent substrates for this pattern, educators can explore solutions interactively on the Pareto front. In practice, this looks like a dashboard where teachers have sliders for objective weights, what-if panels to simulate the consequences of a choice, and "lock/override" controls to fix certain assignments or groups before the GA refines the rest.

### Interaction patterns that work

HITL design borrows proven motifs from interactive machine learning and interactive evolutionary computation (IEC), where human evaluators rate candidate solutions and directly guide the evolutionary search. Some main principles:

- *preference-guided evolution.* In IEC (Takagi, 2001), users iteratively rate or select preferred candidates; the evolutionary loop learns a surrogate of human preference and explores accordingly. For education, this means teachers can quickly mark acceptable groupings or timetables; the GA then extrapolates teachers' intent from what they have already marked to generate new, better potential options. Pairing this with Bayesian preference learning reduces fatigue (Brochu, de Freitas & Ghosh, 2010).

- *transparent explanations.* Educators need to understand the motivation behind a suggestion. Model-agnostic tools provide explanations that, also in education, translate a model's input and output into meaningful narratives. They include SHAP to attribute features (Lundberg & Lee, 2017), counterfactuals to show "what would need to change" (Wachter, Mittelstadt & Russell, 2017), and recourse to surface actionable steps (Ustun, Spangher & Liu, 2019). "What-If" tooling (Wexler et al., 2019) lets staff inspect decision boundaries, probe subgroup performance, and test hypothetical learners before deploying rules at scale.
- *authoring and override.* Teachers must be able to author rules and content, not only react to them. Authoring environments like CTAT, Cognitive Tutor Authoring Tools (Aleven et al., 2006), allow non-programmers to both build and audit ITS behaviors: analogous affordances can be seen in GA-driven systems. Every automated recommendation should be paired with an override with reason flow; captured justifications then seed the next generation of evolution.
- *visual analytics for evolution.* Visual encodings of populations (Llorà et al., 2006), fitness landscapes, and rule evolution (Sreenath & Jeyakumar, 2016) help instructors meaningfully guide the search. Simple patterns like, "pin" a candidate, "ban" a pattern, "nudge" toward a feature, translate expert intent into evolutionary pressure.

## Teacher augmentation in concrete workflows

Three plausible scenarios in which teachers can be involved in activities and tasks that are carried out in education through GA follow.

- *Early-warning and advising.* A GA outer loop tunes a risk model with configurable window length, thresholds, and fairness constraints. Teachers see explanations (key signals, counterfactuals), confidence, and recommended actions for each student. They can snooze, escalate, or dismiss proposed solutions with rationale; those signals feed back into the evolutionary selection pressure, improving triage over time.
- *Group formation for projects.* Instructors set high-level objectives in group creation that balance prior achievement, diversify schedules, and promote collaboration skills. After the multi-objective evolutionary algorithm runs, they review 3-5 non-dominated lists giving explanations: "diversifies daytime availability"; "maximizes complementary knowledge by components", and so on. The ability to lock pairs and re-evolve the rest preserves teacher autonomy.
- *Sequencing and content routing.* Teachers prepare teaching materials and define constraints, like "no more than one difficult quiz per week". An EC planner proposes several viable sequences with predicted cognitive load and completion lift. Educators pick one, tweak a few nodes (if necessary), and deploy the path to students; the evolutionary engine learns from those edits.

## From co-design to sustained practice

Human-centered EC benefits from co-design with teachers and students, ideally through design-based research cycles that iterate on classroom realities (Anderson & Shattuck, 2012; McKenney & Reeves, 2021). Documentation can help: model cards and datasheets for datasets (Mitchell et al., 2019; Gebru et al., 2021) create a shared language among developers, instructors, and administrators for intended use, limitations, and monitoring plans. Because adoption depends on workload and trust, evaluations should routinely include usability and cognitive-load measures – e.g., SUS (Brooke, 1996), NASA-TLX (Hart, 2006) – alongside accuracy and fairness metrics. Finally, institutional policy should formalize human oversight in actions like escalation pathways, appeal rights, and periodic audits, aligning with the EU AI Act's transparency, logging, and human-control provisions for high-risk systems (European Union, 2024).

## Guardrails against failure modes

Automation bias and over-reliance are real risks: users may overweight algorithmic outputs even when they conflict with context (Parasuraman & Riley, 1997; Bahner, Hupper & Manzey, 2008). Mitigations include calibrated uncertainty, contrastive explanations ("why A over B"), and disagreement highlighting, which draws attention when the model contradicts recent teacher inputs (Miller, 2019). Another hazard is deskilling: if systems always make the first move, instructional craftsmanship weakens, delegating decisions to automated systems. Co-pilot patterns that propose but never preempt, plus periodic "human-only" drills, keep skills up to date. Finally, continual monitoring for concept drift and equity impacts is non-negotiable; fairness terms must be embedded in the fitness function rather than a posteriori (Hardt, Price & Srebro, 2016; Zafar et al., 2017).

## Evaluation and evidence

Evolutionary systems that incorporate HITL processes should be evaluated beyond predictive performance. Recommended practice is a mixed-methods evaluation that combines: (a) model metrics (AUC, calibration, subgroup parity), (b) human factors (teachers' commitment; usability), (c) process analytics (time-to-decision; rate of overrides; rationale diversity), and (d) learner outcomes (achievement, engagement, persistence). When feasible, run lightweight A/Bs or stepped-wedge deployments to estimate causal effects.

A practical HITL checklist for EC in education follows.

1. *Make oversight real.* Ship override with rationale, escalation, and audit logs from day one (European Union, 2024).
2. *Constrain and expose objectives.* Let educators set and see the multi-objective trade-offs (Deb et al., 2002; Zhang & Li, 2007).

3. *Explain with actionability.* Pair SHAP with counterfactuals and explicit recourse (Lundberg & Lee, 2017; Wachter, Mittelstadt & Russell, 2017; Ustun, Spangher & Liu, 2019).
4. *Close the loop.* Capture human feedback as evolutionary pressure (Takagi, 2001; Llorà et al., 2006).
5. *Document and monitor.* Publish model cards/datasheets; track accuracy, equity, workload, and drift (Mitchell et al., 2019; Gebru et al., 2021; Hart, 2006).
6. *Co-design and train.* Allocate budget for teacher co-design and support professional development (Anderson & Shattuck, 2012; McKenney & Reeves, 2021).

## 16.5 Neuroevolution with foundation models (LLMs, multimodal systems) for instruction, assessment, and feedback

Genetic Algorithms (GAs) and broader Evolutionary Computation (EC) are intersecting with large foundation models – Large Language Models (LLMs) and multimodal models – in education to create adaptive instructional agents that can draft materials, assess work, and provide feedback while being optimized for local pedagogical goals (Yang et al., 2024; Guo et al., 2024; Fernando et al., 2023). EC contributes three complementary capabilities here: (i) prompt/program search (evolving prompts, tool-use chains, or planner parameters), (ii) policy search (optimizing how and when an agent acts, asks for clarification, or defers to a human), and (iii) multi-objective control (balancing accuracy, latency, fairness, explainability, and workload). Recent results show that evolutionary strategies can outperform manual prompt engineering (Guo et al., 2024).

### Prompt and tool-chain evolution

Evolutionary prompt optimizers such as Promptbreeder and EvoPrompt treat the LLM as a black-box evaluator and iteratively mutate/crossover candidate prompts (or tool-use plans) to improve downstream task performance (Fernando et al., 2023; Guo et al., 2024). In education, the same pattern can tune rubric-aligned grading, Socratic questioning styles, or feedback tones for specific cohorts/courses. Because curricula, cohorts, and institutional policies drift over time, Pareto-based EC (e.g., NSGA-II, MOEA/D) can keep multiple non-dominated policies available, so educators can select contextually appropriate operating points (Deb et al., 2002; Zhang & Li, 2007).

In Figure 16.1, we propose an LLM-assisted workflow for evolving prompts/policies with evaluation and adoption.

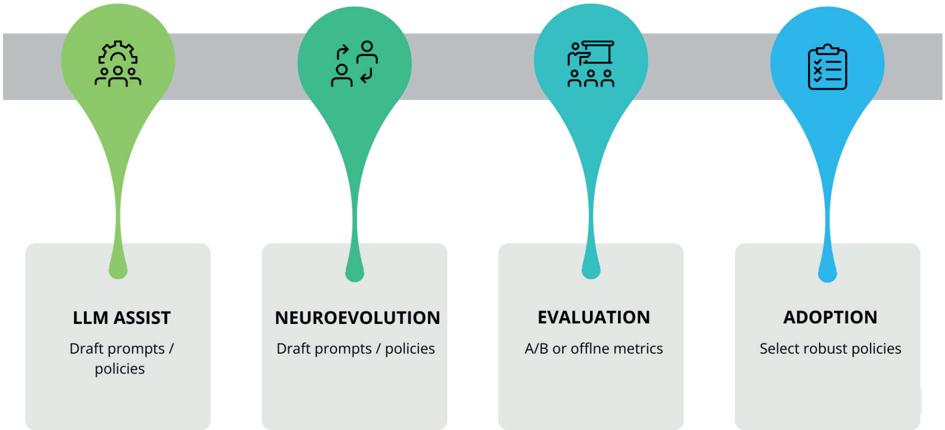


**Figure 16.1** - LLM-Assisted Neuroevolution workflow for educational tutoring

Schematic workflow illustrating how Genetic Algorithms and other evolutionary methods interact with Large Language Models (LLMs) to evolve tutoring prompts and instructional policies. The diagram shows iterative stages of generation, evaluation, and adoption – where LLMs propose or refine prompts and the evolutionary outer loop optimizes them for accuracy, fairness, workload, and pedagogical quality – culminating in human review and institutional deployment.

## Assessment, hints, and formative feedback

LLMs fine-tuned for educational tasks (e.g., rubric-based grading, hint generation) still suffer from brittleness and hallucinations. EC helps by constraining the search with domain rules and by penalizing failure modes directly in the fitness (e.g., truthfulness checks against reference solutions; consistency with worked examples). Mixed-initiative designs – where the model proposes multiple graded explanations and the GA re-ranks/filters them with external verifiers – yield higher trust and alignment to instructor intent. When multimodal inputs (code, diagrams, handwriting) are involved, neuroevolution can tune encoders/decoders or retrieval settings jointly with textual prompts to stabilize accuracy under workload and latency budgets.

## Affect-aware learner support

Learners' emotional states modulate learning gains; modern sensors and log data make it feasible to estimate affective conditions and self-regulation online (D'Mello & Graesser, 2012). EC can evolve policy ensembles that condition LLM-generated feedback on affective cues (frustration, disengagement), optimizing for performance and persistence. In practice, objectives include calibration (avoid overconfident yet wrong answers), helpfulness (reduce time-on-task without sacrificing learning), and equity (parity in false negatives for at-risk detection).

### Governance and safety

HITL oversight, logging, and transparency are not optional. Design constraints are required, as LLM-enabled assessors and advisors are high-risk systems (e.g., learner profiling, automated scoring) under the EU AI Act (European Union, 2024). Every deployed hybrid model should be paired with its documentation artifacts, corresponding to model cards and datasheets (Mitchell et al., 2019; Gebru et al., 2021).

## 16.6 Privacy-preserving and federated evolutionary learning at scale

Data minimization, locality, and accuracy become equally important as institutions adopt analytics across programs and partners. When data cannot leave the campus, department, or country, evolutionary computation can be deployed through federated learning (FL) and privacy-enhancing technologies. Two relevant patterns that avoid transferring raw data and sharing model updates are (i) cross-silo FL, where universities, schools, or other entities coordinate to train shared models without centralizing data, and (ii) cross-device FL, where devices owned by learners perform lightweight local training (McMahan et al., 2017; Kairouz et al., 2021).

### Federated EC: islands, but for privacy

Island-model evolution maps naturally to FL. In such a setting, each institution at the system or course level maintains a local subpopulation; periodic secure migration shares model deltas, candidate genomes, or summary fitness, not raw data. Secure aggregation protocols and differential privacy (DP) noise guard against reconstruction or membership inference (Dwork & Roth, 2014; McMahan et al., 2017). In educational deployments, the “genome” may encode risk-model thresholds, cost weights, or sequencing rules; fitness combines locally computed measures such as AUC/PR, fairness gaps, and advisor workload. A central coordination body merges elites with privacy guarantees, satisfying GDPR’s data minimization and purpose limitation while retaining cross-institution learning benefits (Regulation (EU) 2016/679).

### Private-by-design fitness and evaluation

Even when data remain local, fitness signals can leak. Practical mitigations include (a) DP mechanisms on metrics (e.g., adding calibrated noise to AUC/TPR/FNR), (b) clip-and-noise for gradient- or embedding-sharing pipelines, and (c) secure enclaves for high-sensitivity evaluations (Dwork & Roth, 2014; Kairouz et al., 2021). In multi-objective EC, it is recommended to add privacy budget consumption as an explicit objective or penalty; this surfaces the Pareto trade-off between accuracy and privacy, so decision makers can choose acceptable operating points (Deb et al., 2002).

### When LLMs meet privacy constraints

If LLMs are part of the loop (e.g., for feedback authoring), prefer on-prem or institutionally hosted models for sensitive data, or deploy retrieval-augmented generation (RAG) where only de-identified embeddings are shared. Under the EU AI Act, providers and deployers must implement logging, risk management, and human oversight for high-risk systems; countries collaborating cross-border need to align their privacy and AI governance (European Union, 2024). Where bandwidth and cost bind, consider federated prompt evolution: institutions evolve prompts locally and share only anonymized prompt templates and aggregate payoffs, without student data.

### Equity under privacy

DP can disproportionately harm the utility of minority or disadvantaged groups due to noise; conversely, privacy can act as a shield protecting vulnerable learners from profiling (Dwork & Roth, 2014; Hardt, Price & Srebro, 2016). Treat fairness as a first-class objective alongside privacy and accuracy. Techniques such as equalized-odds constraints and per-group calibration can be embedded directly in the fitness to counteract disparate impact, even in federated settings (Hardt, Price & Srebro, 2016; Zafar et al., 2017).

## 16.7 Operationalizing responsible EC/ML in education (MLOps, monitoring, and governance)

Turning prototypes into reliable services demands operations as much as algorithms. Educational teams need pipelines that make EC repeatable, auditable, and aligned with institutional policy.

Figure 16.2 shows a layered MLOPs architecture tailored to Evolutionary Computation (EC) workflows in education.

### Pipelines and drift-aware retraining

Because cohorts, curricula, and platforms change, concept drift is the norm (Gama et al., 2014). A robust stack pairs prequential evaluation (test-then-train) with scheduled light-touch evolutionary updates of only the most drift-sensitive knobs (e.g., class weights, thresholds, windows). Continuous integration applies to data and genomes: pin dataset versions (Caliper/xAPI event schemas), fix random seeds, and track genomes and fitness histories for reproducibility (Advanced Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; 1EdTech Consortium, 2020). Many failures in production come from glue code and mismatch across systems, not the model; technical debt management and clear ownership are key (Sculley et al., 2015).

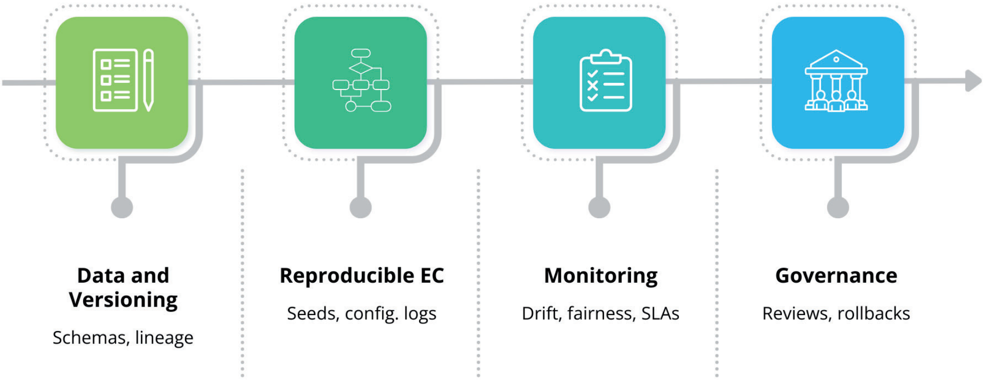


**Figure 16.2** - MLOps Stack for Evolutionary Computation in education
Layered architecture illustrating a responsible MLOps environment tailored to Evolutionary Computation (EC) workflows in education. The figure depicts interconnected modules for data and version management, reproducible EC runs, drift-aware retraining, monitoring, and governance. It emphasizes traceability – from dataset versioning and genome tracking to fairness dashboards and compliance logging – ensuring that optimization pipelines remain transparent, auditable, and aligned with institutional and regulatory standards.

### Interoperability and data standards

Caliper and xAPI provide portable event vocabularies and transport protocols for learning traces, enabling reproducible EC pipelines across vendors and institutions (Advance Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; 1EdTech Consortium, 2020). For optimization tasks like timetabling, open benchmarks (e.g., ITC/EJOR educational timetabling datasets) and public leaderboards help drive comparability and innovation (Ceschia, Di Gaspero & Schaerf, 2023; Bashab et al., 2023).

### Human-centered operations

As already emphasized, keep educators in the loop. Operationalize override with rationale, appeal flows, and explanation views (Lundberg & Lee, 2017). Document and monitor workload effects, not only accuracy (Hart, 2006). Create governance boards that include faculty, students, and accessibility experts to approve objective functions and review drift/fairness reports (Slade & Prinsloo, 2013; Drachsler & Greller, 2016).

### Documentation and auditability

Model cards and datasheets clarify intended use, limits, and evaluation (Mitchell et al., 2019; Gebru et al., 2021). For high-risk analytics (profiling, automated scoring), the EU AI Act requires risk management, data governance, logging, transparency, human oversight, accuracy/robustness, and post-market monitoring; educational deployers should maintain audit logs of training/evolution runs, parameter changes, override events, and appeals (European Union, 2024).

### Evaluation beyond accuracy

Multi-objective EC already supports operational criteria (latency, compute cost, advisor workload) and ethical criteria (calibration, subgroup parity, equalized-odds gaps). It is suggested to adopt standardized dashboards that report: (a) model metrics (AUC/PR, calibration), (b) fairness diagnostics across legally/protected-relevant groups, (c) human factors (adviser time, override rates), and (d) cohort drift signals. Ablations of fairness/latency budgets to make the trade-offs concrete should be included in publication or internal reviews.

### Sustainability and cost

Compute is not free, either financially or environmentally. Budget-aware objectives (GPU-minutes, energy use) and green-AI practices (early stopping, multi-fidelity evaluation, smaller populations) should be incorporated into the fitness function (Schwartz et al., 2020; Jin, 2005, 2011; Strubell, Ganesh & McCallum, 2019).

## 16.8 Open challenges

In the last pages, we have faced many topics on the applications of EC in education processes: from the integration of Genetic Algorithms with AI, big data, and LLMs, to MLOps practices, the role of humans in decision-making in automated systems, issues of privacy and fairness, as well as scalability and the adoption of technological solutions like FL in institutional settings. We reviewed applications of EC in formal contexts and proposed some for workforce training.

Looking ahead, we list the open challenges to translate current advances in evolutionary systems into robust, real-world educational deployments.

### Causal and longitudinal evaluation

Many promising EC deployments optimize short-term proxies (next-week risk, immediate completion) rather than long-term learning, transfer, or well-being. The field needs more randomized and quasi-experimental designs (Shadish, Cook & Campbell, 2002) that track multi-term outcomes, retention, and equity impacts, especially for at-risk students. Embedding experimental toggles in EC (e.g., evolving when to intervene, not whether) can reveal causal levers.

### Benchmarks, open data, and standards

OULAD catalyzed early-warning research by providing a well-documented, de-identified corpus (Kuzilek, Hlosta & Zdrahal, 2017). For timetabling and allocation, sustaining public benchmarks (ITC/EJOR) keep algorithmic progress measurable (Ceschia, Di Gaspero & Schaerf, 2023). The community needs updated, privacy-preserving multi-institution corpora, standard instances, plus task suites for (a) fair risk modeling, (b) preference-aware timetabling, (c) grouping, and (d) curriculum sequencing. On the ed-tech operations side, shared schemas via Caliper/xAPI reduce bespoke Extract, Transform, Load (ETL) pipelines (Advanced Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; 1EdTech Consortium, 2020).

### Human factors and teacher augmentation

Robust evidence on workload, trust, and skill development is thin. Studies should evaluate interactive evolutionary computation and the HITL approach to EC along human-centered metrics and measure deskilling/skill-building effects over time. Design science cycles with co-creation by teachers and students can surface tacit constraints that purely data-driven optimization overlooks (Anderson & Shattuck, 2012; McKenney & Reeves, 2021).

### Fairness, privacy, and inclusivity by design

Equalized odds and related constraints should move from post-hoc auditing into optimization objectives (Hardt, Price & Srebro, 2016). Likewise, DP budgets and auditability should be encoded in the genome/fitness, not bolted on, and evaluated under FL to ensure equity under privacy (Dwork & Roth, 2014; Kairouz et al., 2021). Accessible-by-default designs and disability-aware objectives need dedicated benchmarks and metrics.

### Sustainability and cost-aware optimization

Green-AI metrics and cost ceilings should be standard in fitness, especially for resource-constrained institutions (Schwartz et al., 2020; Strubell, Ganesh & McCallum, 2019). Surrogate-assisted and multi-fidelity EC can sharply reduce evaluations without degrading outcomes (Jin, 2005, 2011).

### Neuroevolution with LLMs under constraints

We lack rigorous methods to evolve LLM-based tutors that are grounded (reliable), private, and fair. Research should compare evolutionary prompt/program search versus RL-based approaches under explicit EU AI Act constraints (logging, human oversight) and GDPR requirements (consent, minimization, DP) (European Union, 2024; Regulation (EU) 2016/679). Promising directions include: (i) federated prompt evolution, (ii) privacy- and fairness-aware fitness, and (iii) explanation-aware fitness that rewards faithful rationales (Yang et al., 2024; Guo et al., 2024; Mitchell et al., 2019).

### From prediction to prescription

Much EDM focuses on predicting risk. Many studies present post-hoc analyses that remain experimental rather than being applied in real institutional settings. EC can enable prescriptive analytics, optimizing what to do (message timing/type, content sequence, group composition) under constraints (adviser time, fairness) in real time. Field experiments that connect prescriptions to outcomes – and report null results – will mature the evidence base (Papamitsiou & Economides, 2014; Ferguson, 2012).

# Chapter 17

## Conclusions

This book set out to examine how Genetic Algorithms (GAs) and Evolutionary Computation (EC) can be harnessed within Learning Analytics (LA) and Educational Data Mining (EDM) to optimize educational systems, meet multifaceted institutional goals, and foster personalization and inclusivity. Across chapters, we showed that EC's population-based search, multi-objective framing, and iterative refinement make it an unusually good fit for the messy realities of education, including non-linear trade-offs, heterogeneous cohorts, non-stationary data, and resource constraints that shift by course, term, and institution. In doing so, we moved beyond narrow "accuracy-at-all-costs" perspectives to a broader, practice-ready view that integrates effectiveness, equity, privacy, explainability, and operational feasibility.

### What this work demonstrated

First, GAs and EC are particularly strong when problems are inherently combinatorial, multi-criteria, or dynamic (see Figure 17.1). We saw this in curriculum sequencing and timetabling, where student preferences, instructor constraints, rooms, time zones, and fairness considerations meet; matheuristic approaches pairing GA

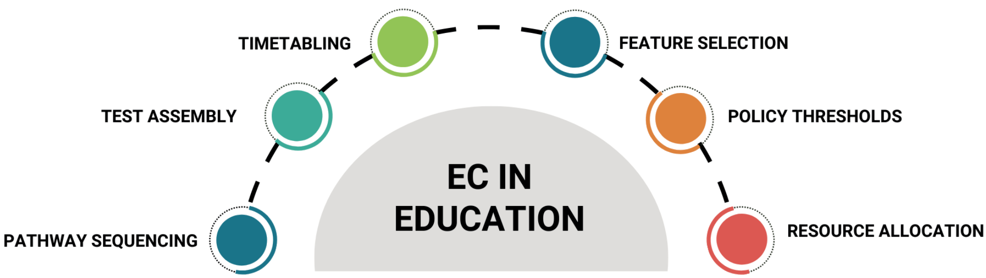


**Figure 17.1** - Mindmap of Evolutionary Computation across the educational pipeline

Conceptual diagram summarizing where Evolutionary Computation (EC) enhances educational processes: curriculum sequencing, test assembly, timetabling, feature selection, threshold tuning, and resource allocation. It highlights EC's integrative role in optimizing complex, multi-criteria, and dynamic problems across learning analytics and institutional operations.

search with exact repair routinely achieve high-quality schedules at the institutional scale (Elshani & Pireva Nuçi, 2021; Hssina & Erritali, 2019; Christudas, Kirubakaran & Thangaiah, 2018; Nogueira et al., 2024; Ceschia, Di Gaspero & Schaerf, 2023; Mahlous & Mahlous, 2023; Puchinger & Raidl, 2005; Dunke & Nickel, 2023). We saw it in predictive analytics where feature selection, cost-sensitive thresholds, and calibration must be optimized together to support early-warning pipelines without overwhelming advisors (Tan et al., 2017; Xue et al., 2016; Queiroga et al., 2020; Hao, Gan & Zhu, 2022; Sahlaoui et al., 2024). And we saw it in collaborative learning and group formation, where GAs can encode richer notions of diversity and complementarity than ad-hoc heuristics, improving both outcomes and perceived fairness (Moreno, Ovalle & Vicari, 2012; Li, Ouyang & Chen, 2022).

Second, hybridization matters. EC's global search and constraint handling combine naturally with modern machine learning, from tree ensembles to deep networks. This leads to a reasonable, practical division of labor: learning systems extract signal and represent structure; EC discovers feature subsets, hyperparameters, sequencing rules, and operating points that satisfy multiple goals (Ouyang, Zheng & Jiao, 2022; Charitopoulos, Rangoussi & Koulouriotis, 2020; Xue et al., 2016; Queiroga et al., 2020; Zakaria, Anas & Oucamah, 2022; Sahlaoui et al., 2024). Surrogate-assisted and multi-fidelity evolution lower costs (Jin, 2005, 2011) while maintaining competitive accuracy, converting what can be a research-only technique into a production-ready one. The development of tools such as KEEL and jMetalPy made it straightforward to compare evolutionary methods, run controlled experiments, and parallelize evaluations, which are key conditions for reproducibility and scale (Alcalá-Fdez et al., 2009; Benítez-Hidalgo et al., 2019).

Thirdly, the domain is moving from prediction to prescription. EC goes beyond "who is at risk" to discover "what to do about it". This means choosing a message strategy, sequencing a remedial pathway, composing different groups, or shifting the exam calendar, under explicit constraints for teacher/tutor workload, latency, privacy, and equity. Multi-objective algorithms (e.g., NSGA-II, MOEA/D) provide transparent Pareto fronts, so that academic boards can select operating points that reflect institutional values and legal obligations (Deb et al., 2002; Zhang & Li, 2007; Hardt, Price & Srebro, 2016; Zafar et al., 2017).

## Integrations that unlock new capabilities

A major strand of this work explored the connections between EC and big data infrastructures and deep learning, and the most recent foundation models. Distributed and cloud-native designs make evolutionary search tractable on VLE-scale corpora like OULAD, where the volume and velocity of clickstream, assessment, and forum data would otherwise overwhelm serial pipelines (Kuzilek, Hlosta & Zdrahal, 2017; Waheed et al., 2020; Reich & Ruipérez-Valiente, 2019). Federated patterns (McMahan et al., 2017; Kairouz et al., 2021) extend these gains to settings that can manage privacy issues: island models adjust cleanly onto cross-silo federated learning, letting institutions exchange model deltas or fitness summaries rather than raw data, while differential privacy adds (noise and) formal protections (Dwork & Roth, 2014). With the diffusion of Large Language Models and multimodal systems also in teaching and learning activities, neuroevolution's role has expanded from hyperparameters to prompt/program search, tool-use policies, and multi-objective guardrails (truthfulness, workload, fairness), making LLM-based tutors and assessors more aligned to local pedagogy and policy (Yang et al., 2024; Guo et al., 2024; Liu et al., 2024; OpenAI, 2023; Kasneci et al., 2023).

## Equity, privacy, and accountability as design goals

Throughout, we treated ethics not as an afterthought but as a first-class objective. That decision changes the engineering. We have proposed to encode equalized-odds gaps, stability of explanations, and advisor workload directly into the fitness; we have suggested to adopt prequential drift monitoring to prevent silent degradation across cohorts; and we have borrowed data minimization, pseudonymization, DP budgets from privacy engineering to keep optimization safe by design (Hardt, Price & Srebro, 2016; Zafar et al., 2017; Gama et al., 2014; Advanced Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; 1EdTech Consortium, 2020; Dwork & Roth, 2014). We have also underlined the necessity to recognize the regulatory moment and align with Regulation (EU) 2016/679 (GDPR) and Regulation (EU) 2024/1689 (AI Act), in which learner-profiling and automated-scoring tools qualify as high-risk systems, triggering requirements for risk management, data governance, transparency, human oversight, robustness, and post-market monitoring.

## Implications for practice

For educators, evolved rules and visual explanations turn black boxes into cooperative and supportive work devices. Because EC can optimize for interpretability (e.g., rule lists, sparse subsets) alongside accuracy, faculty can interrogate and override suggestions with confidence, keeping human judgment in the loop (Lundberg & Lee, 2017; Takagi, 2001). For program leaders, multi-objective solutions (accuracy/latency/fairness/workload) align analytic choices with policy, resource, and equity goals, e.g., limiting weekly alert volume, guaranteeing parity across demographics, or capping GPU-minutes. For IT-strategy managers and platform teams, containerized, parallel evolutionary pipelines, event standards (Caliper, xAPI), and experiment tracking reduce technical debt and make results replicable across semesters and departments (1EdTech Consortium, 2020; Advanced Distributed Learning Initiative, 2016; IEEE Standard Association, 2023; Sculley et al., 2015). And for students, the practical outcome is quieter but consequential: timetables that respect constraints and preferences; groupings that feel fair and well-equipped; interventions that are timely, humane, and effective; and learning pathways that adapt without stigmatizing. Figure 17.2 describes how optimization-as-design can serve different stakeholders in education.

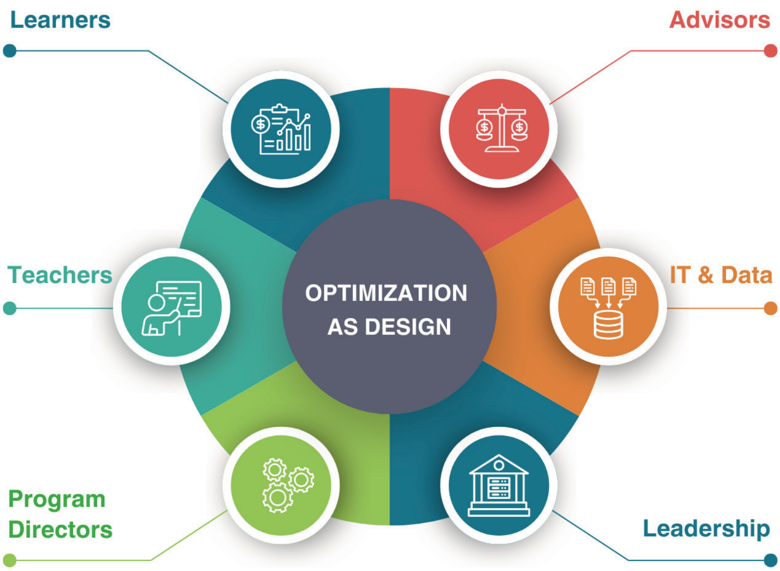


**Figure 17.2** - Stakeholder compass for Optimization-as-Design in education

Diagram illustrating how optimization-as-design serves multiple educational stakeholders: students, teachers, advisors, program directors, IT/data teams, and institutional leadership. It highlights the shared benefits of evolutionary optimization, from personalized learning and fair resource allocation to transparent governance and scalable, data-driven operations.

## Limitations and cautions

The book has been candid about limits. Compute budgets and carbon cost are real; even with surrogates and multi-fidelity evaluation, careless search spaces can make runtime explode (Jin, 2005, 2011; Schwartz et al., 2020; Strubell, Ganesh & McCallum, 2019). Data quality and representativeness remain the rate-limiters for many institutions; privacy-preserving methods can reduce leakage but also dilute the signal for small groups (Dwork & Roth, 2014). Offline accuracy of post-hoc analyses is not a proxy for educational impact; interventions that look optimal on logs can underperform in practice if they increase advisor burden or erode student trust. And while EC can encode fairness directly, it cannot by itself fix upstream inequities in access, preparation, or support that already exist in the educational system (Hardt, Price & Srebro, 2016; Slade & Prinsloo, 2013). These cautions argue for rigorous governance: documentation (model cards, datasheets), audit trails, HITL (Human-in-the-Loop) review, and participatory design with educators and learners (Mitchell et al., 2019; Gebru et al., 2021; Takagi, 2001; Anderson & Shattuck, 2012; McKenney & Reeves, 2021).

## A practical synthesis

If there is a single lesson from the chapters, it is this: scalable, trustworthy optimization in education emerges from the combination of lean evaluators, disciplined search spaces, parallel/asynchronous execution, and objectives that bake in not only accuracy but also latency, cost, privacy, fairness, interpretability, and workload. This "whole-stack" view turns EC from a lab curiosity into an operational asset for early-warning systems, preference-aware timetabling, adaptive grouping, and curriculum sequencing, also at scale in MOOC platforms or VLEs.

## Where research should go next

Several directions are pressing. First, causal and longitudinal evaluation: field experiments in which evolved prescriptions – what to do and when – are linked to learning, transfer, wellbeing, and equity outcomes in real contexts over terms, not just weeks. Second, privacy- and fairness-aware neuroevolution for LLM-enabled tutors and assessors: federated prompt/policy evolution, explanation-aware fitness, and explicit governance aligned to EU-AI-Act. Third, shared benchmarks and standards: updated, de-identified multi-institution corpora, open timetabling/grouping/sequencing suites, and greater adoption of Caliper/xAPI for reproducible pipelines. Fourth, sustainability: budget-aware objectives and green-AI practices to ensure access by resource-constrained institutions. Finally, human factors: systematic study of HITL processes and workload, trust, and skill-building for teachers working with EC-augmented tools.

# Chapter 18

# Closing remarks

Education forces us to hold competing goods together: rigor and care, efficiency and inclusion, privacy and personalization. One of the central claims developed across this book is that Evolutionary Computation (EC) and, in particular, Genetic Algorithms (GAs) provide a language and a toolset for examining the complexity of educational settings and balancing those goods instead of reducing them to a single objective. EC makes trade-offs explicit via Pareto front reasoning and searches over diverse candidate solutions; this is one of the reasons it is unusually well-suited to education's nonlinear objectives, heterogeneous cohorts, non-stationary data, and shifting resource constraints. It manages to simplify lengthy procedures while still saving flexibility in choosing the course of action to be taken.

Here EC is not just about optimization; it's about steering complex educational systems through uncertainty and safeguarding values.

## 18.1 From optimization to orchestration

Across the volume's two arcs – fundamentals of EC (Chs. 1-9) and its deployment in learning analytics and educational data mining (Chs. 10-17) – some patterns emerge:

- *multiple legitimate goals must be pursued simultaneously* (e.g., accuracy and advisor workload; throughput and fairness). Through EC's multi-objective framing, tensions among goals are viewed as first-class design elements rather than secondary considerations (cf. Ch. 7 on multi-objective evolutionary strategies).
- *search over structural matters.* In education, we often need to optimize combinatorial objects (timetables, curriculum sequences, student groups, resource allocations). EC's population based search, hybridized with exact "repair" or constraint-handling, delivers high-quality feasible plans (examples and references in Chs. 11-12).
- *adaptation over time is the norm.* Cohorts, policies, and platforms shift. By integrating monitoring, an iterative evolutionary approach enables continual recalibration in operational settings (main considerations introduced in Chs. 11-14; monitoring and governance emphasized in Chs. 15-16).

## 18.2 What this book demonstrates in practice

Several case types (see Ch. 12) recur, illustrating why EC fits the educational context:

1. *institutional scheduling and timetabling.* A natural tension exists among student preferences, instructor constraints, rooms, time zones, accessibility, and fairness criteria. Matheuristic designs that pair GA search with exact repair consistently achieve high-quality, feasible rosters and schedules under tight constraints.
2. *curriculum sequencing and pathways.* Sequencing must balance, among others, prerequisite mastery, motivation, cognitive load, learning goals, content layouts, and platform constraints. EC's structure search over sequences or policies supports progression that adapts to diverse learners and platform scale.
3. *predictive early-warning and triage pipelines.* Alerts must be timely and actionable, yet not overwhelming to teachers or tutors; this requires jointly optimizing feature selection, calibration, and cost-sensitive thresholds. EC provides end-to-end search over such pipelines, explicitly incorporating institutional capacities.
4. *group formation and collaborative learning.* Dividing students into teams, encoding richer notions of complementarity and diversity than simple heuristics can improve both outcomes and perceived fairness. EC allows such social objectives to co-exist with performance ones.
5. *hybrid ML × EC systems.* EC complements modern ML by searching across model families, feature/hyperparameter sets, training budgets, ensembling strategies, and even decision policies. For text-rich educational signals, combining LLM-aided representation with Genetics-Based Machine Learning (GBML) is highlighted in Ch. 14; neuroevolution is treated with practical guardrails in Section 14.4.

Across these settings, the methodological thread is consistent and follows common steps: explicitly define the competing goods; encode them in the fitness or constraints; and search over policy-level decisions rather than only model weights. These standards provide the foundations to move from predicting outcomes theoretically to designing interventions and structures practically.

## 18.3 Ethics as design, not decoration

A through-line of the book is that ethical criteria must be included from the start in the fitness function and the operational envelope (see Ch. 15 and the summary in Chs. 16-17):

- *fairness* is operationalized via constraints/metrics (e.g., equalized-odds–style gaps) and balanced against utility during search rather than applied post hoc after deployment.
- *privacy* engineering is treated as an algorithmic constraint: data minimization, pseudonymization, and differential-privacy budgets are design inputs, not toggles.

- *monitoring* for drift and harm uses prequential/continual evaluation to prevent performance and disparities from silently degrading over time and across cohorts.
- *human governance* through documentation, auditing, oversight, and co-design with advisors, teachers, students, and accessibility experts is a non-optional layer that shapes what “good” looks like in context.

The upshot is practical: responsibility is engineered, not asserted. When EC is deployed with these constraints, “optimization” becomes a vehicle for learning and value alignment inside the institutions instead of just a technical competition for accuracy.

## 18.4 Patterns for practice: from sandbox to institution

The MLOps and design-pattern perspective introduced in Chapter 14 and extended through Chapters 15-16 suggests several concrete patterns that help EC systems succeed beyond the lab.

1. *Start multi-objective.* Do not prematurely squeeze a multi-objective goal into a single scalar. Define fairness, workload, timeliness, and feasibility as explicit objectives or constraints, determine the Pareto front, and visualize it with stakeholders (Chs. 7; Chs. 16-17).
2. *Constrained by design.* Ensure that candidates respect institutional rules (e.g., timetable opportunity) by using feasibility repair, penalty terms, and constraint-handling operators. This makes evaluation meaningful and reduces post-hoc fixes (Chs. 5-6; Ch. 17).
3. *Hybridize deliberately.* Pair EC with tree ensembles, calibrated probabilistic models, and/or LLM-supported feature extraction, but bound the search space and compute budget. When using neuroevolution, follow the practical guidance and limitations outlined in Ch. 14.
4. *Instrument for governance.* Produce documentation, audit trails, and cohort dashboards into the pipeline. Monitor error/fairness drift prequentially; treat DP budgets as consumable resources (Chs. 15-17).
5. *Co-design and iterate.* Consider advisors, instructors, and even students as co-designers of the project who help you set goals, define harms, and assess acceptability. It is not an option; it is the only mechanism by which external validity and sustainability can be assured (foregrounded in Chs. 15-16).
6. *Plan the scale-up.* Use the roadmap from pilot to institutionalization of EC in education (Figure 18.1) as an action plan: pilot with small cohorts; validate under distribution shift; stage rollouts with governance gates; and invest in maintainability and sustainability.

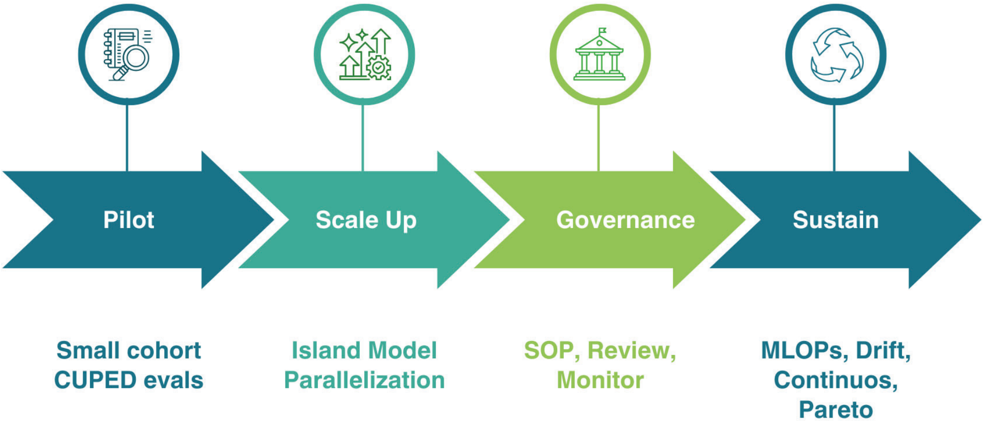


**Figure 18.1** - Roadmap from pilot to institutionalization of EC in education
Timeline diagram outlining the progressive phases of deploying EC/GA solutions in educational institutions from pilot experimentation to staged rollout and full institutionalization. It emphasizes scale-up planning, governance checkpoints, and sustainability milestones to ensure responsible, auditable adoption of optimization-as-design practices.

## 18.5 A research and policy agenda

We have surveyed the main challenges and future research directions in EC applications in education at the end of Chapters 16 and 17; this closing argument benefits from a consolidated and comprehensive agenda.

1. *Causal evaluation at scale*. Move from short-horizon accuracy gains to long-horizon outcomes: this requires field experiments and quasi-experimental designs that connect evolved policies to student trajectories over several months.
2. *Governed personalization with privacy.* Care of fairness and privacy issues in neuroevolution when working on LLM-enabled tutors and assessors, by adopting federated prompt/policy evolution, and explanation-aware fitness, and aligning with emerging regulatory frameworks.
3. *Shared benchmarks and standards.* Create/Adopt open, de-identified multi-institution corpora and problem suites for issues such as timetabling, group formation, and curriculum sequencing with community-defined standards for utility, equity, privacy loss, and operational cost. It is important for incremental progress.
4. *Human-in-the-loop methods.* Formalize procedures and moments when human expertise gets into the loop: preference learning from advisors, value shaping of fitness components, and interactive visualization of Pareto sets to support deliberation.
5. *Robustness under non-stationarity.* Develop evaluation protocols and operators that can adapt to cohort, policy, and platform shift, acknowledging that educational environments evolve alongside the model.

6. *Institutional capacity building.* Work to ensure that policy recognizes that responsible EC/ML is not a one-off procurement and requires ongoing investment in data stewardship, security, accessibility, and continuous monitoring.

## 18.6 What "responsible use" looks like in the wild

We can distill several concrete kinds of decisions that EC can support when used responsibly. A mature institutional EC-enhanced portfolio might include:

- a timetable that works for everyone, trading off instructional priorities with needs, accessibility, and equity;
- a message that is delivered to the recipient when it can still be helpful, calibrated to the advisor's capacity and the student's receptivity;
- a group where every student belongs, with prior knowledge balanced with diversity and complementarity;
- a pathway that adapts to each student without prejudice, revising and explaining recommended content while respecting constraints and privacy;
- a resource allocation that is explainable, where the reason for a match made is as transparent as the match itself;
- a protective monitoring regime, surfacing drift or disparity early, and triggering human review.

In each case, the artifact resulting from a decision within an EC system, such as a schedule, allocation, alert, or recommendation, is the product of a multi-objective, constrained, audited, and co-designed search process. That is the core working idea of this book.

## 18.7 A practitioner's closing checklist

To maximize the usefulness of closing arguments, here is a brief checklist where each item is discussed into book chapters:

1. objectives named and plural (utility, equity, workload, timeliness, feasibility) and reflected in fitness and constraints (Ch. 7; Ch. 12; Ch. 17).
2. feasibility by design (repair operators, constraint-handling) for combinatorial artifacts like timetables (Chs. 5-6; Ch. 17).
3. hybridization plan including which ML components are used, what is evolved, budget bounds, and failure modes (Ch. 14).
4. governance instrumentation comprising documentation, audit trails, cohort-wise dashboards, and prequential drift monitoring (Chs. 15-17).

5. privacy posture with data minimization, pseudonymization, and DP budgets treated as first-class resources (Chs. 15-17).
6. fairness posture governing model performance and human impact and established in metrics, objectives, and constraints (equalized-odds; explanation stability; workload bounds) (Ch. 17).
7. human-in-the-loop processes realized through co-design of objectives, stakeholder review of Pareto sets, and escalation policies (Chs. 15-16).
8. scale-up roadmap from pilot to staged rollout to institutionalization, with gates and sustainability plans (Figure 18.1).

For an effective institutional deployment, any of the previous points has to be checked.

## 18.8 Final perspective

The optimism of this conclusion is deliberate but not naïve. EC is not a silver bullet: it is a disciplined way to negotiate trade-offs transparently. Its value in education lies in making choices and consequences transparent, contestable, and improvable over time. When objectives are plural, constraints are real, and values are non-negotiable, EC helps us search the space of what is possible rather than settle for what is easy.

Used responsibly (namely, documented, audited, overseen, and co-designed), EC can assist institutions in transforming noisy traces into actionable and equitable decisions. Balancing technique and governance, the approaches across this book can support a future of learning that is more personal, more fair, and more effective for *every learner*.

# Appendix A

## GeneticaR

*geneticaR* is the companion R package for this book. It implements single- and multi-objective genetic algorithms (including NSGA-II), supports binary/real/permutation encodings, offers penalty- and repair-oriented constraint handling, plotting, and parallel evaluation, and ships with reproducible scripts tailored to Learning Analytics scenarios. The package is meant to make the methodological ideas of the book concrete and auditable in code, including the “GA ↔ LLM” workflows discussed in Part I.

This book argues that many real educational design problems – timetabling, curriculum sequencing, feature/rule selection, policy optimization under fairness and workload constraints – are naturally treated as multi-criteria search over jagged, mixed spaces. Genetic Algorithms (GAs) provide a principled, auditable way to surface trade-offs (Pareto fronts) and to balance constraints with outcomes. *geneticaR* operationalizes these ideas in idiomatic R so that readers can reproduce the figures and extend the examples in their own institutions.

Scope at a glance (see also Table 5.3 in the book): single- and multi-objective GAs (NSGA-II), selection schemes (tournament, roulette, rank, SUS), crossover (1/2-point, uniform, BLX-α, SBX, PMX), mutation (bit-flip, Gaussian, polynomial, swap), penalty-based constraints and hooks for repairs, parallel evaluation, and plotting utilities.

A curated repository hosts the package source (geneticaR/), example scripts (/scripts), and small datasets for the book’s worked examples (/datasets).

Visit the Zenodo project and download the geneticaR package.

**Repository:** http://bit.ly/43MhO8J

**DOI: 10.5281/zenodo.17683598**

A complete documentation is in /docs.

Installation from source (R ≥ 4.1):

```
# install.packages("remotes")
remotes::install_local("path/to/geneticaR", build_vignettes = TRUE)
# or devtools::install_local("...") if you prefer devtools
```

After installation, consult the vignette:

```
browseVignettes("geneticaR")
```

GAs orchestrate a population of candidate solutions that undergo selection, crossover, and mutation until termination. NSGA-II augments this with non-dominated sorting and crowding distance to approximate a Pareto set. The book's Figures and Tables (e.g., the GA flow in Ch. 3; encoding table; toolkit overview) provide the conceptual background that geneticaR mirrors in code.

# References

1EdTech Consortium (2020). Caliper Analytics® Specification (Version 1.2, Final Release). https://www.imsglobal.org/spec/caliper/v1p2

Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

Abdelhalim, E., & El Khayat, G. (2016). A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA). *Alexandria Engineering Journal*, *55*(2), 1395-1409. https://doi.org/10.1016/j.aej.2016.02.017

Abdipoor, S., Yaakob, R., Goh, S. L., & Abdullah, S. (2023). Meta-heuristic approaches for the university course timetabling problem. *Intelligent Systems with Applications*, *19*, 200253. https://doi.org/10.1016/j.iswa.2023.200253

Advanced Distributed Learning (ADL) Initiative (2016). Experience API (xAPI) specification (Version 1.0.3) [Specification]. GitHub. https://github.com/adlnet/xAPI-Spec

Agrawal, A., Agarwal, A., & Bansal, P. (2021). Integration of fuzzy logic with Metaheuristics for education center site selection. *Education and Information Technologies*, *26*(1), 103-124. https://doi.org/10.1007/s10639-020-10254-9

Al Farissi, A., Dahlan, H. M., & Samsuryadi. (2020). Genetic Algorithm Based Feature Selection for Predicting Student's Academic Performance. In F. Saeed, F. Mohammed, & N. Gazem (Eds.), *Emerging Trends in Intelligent Computing and Informatics. IRICT 2019. Advances in Intelligent Systems and Computing*, vol 1073 (pp. 110-117). Springer, Cham. https://doi.org/10.1007/978-3-030-33582-3_11

Alba, E., & Tomassini, M. (2002). Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *6*(5), 443–462. https://doi.org/10.1109/TEVC.2002.800880

Albreiki, B., Zaki, N., & Alashwal, H. (2021). A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques. *Education Sciences*, *11*(9), 552. https://doi.org/10.3390/educsci11090552

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., Garcia, S., Sanchez, L., & Herrera, F. (2011). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of multiple-valued logic and soft computing*, 17, 255-287.

Alcalá-Fdez, J., Sánchez, L., García, S., del Jesús, M. J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., & Herrera, F. (2009). KEEL: A software tool to assess evolutionary algorithms for data-mining problems. *Soft Computing*, *13*(3), 307–318. https://doi.org/10.1007/s00500-008-0323-y

Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R. (2006). The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. In M. Ikeda, K.D. Ashley, & TW. Chan (Eds.) I*ntelligent Tutoring Systems. ITS 2006. Lecture Notes in Computer Science*, vol 4053 (pp. 61–70). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11774303_7

Alias, M. H., Hambali, N., Aziz, M. A., Taib, M. N., & Jailani, R. (2024). Feature selection techniques and classification algorithms for student performance classification: a review. *International Journal of Electrical and Computer Engineering*, *14*(3), 3230-3243. https://doi.org/10.11591/ijece.v14i3.pp3230-3243

Allwein, E. L., Schapire, R. E., & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, 113–141. http://www.jmlr.org/papers/v1/allwein00a.html

Anderson, T., & Shattuck, J. (2012). Design-Based Research: A Decade of Progress in Education Research?, *Educational Researcher*, *41*(1), 16-25. https://doi.org/10.3102/0013189X11428813

Ary, D., Jacobs, L.C., Irvine, C.K.S., & Walker, D. (2010). *Introduction to Research in Education* (8th ed.). Belmont (CA): Wadsworth Cengage Learning.

Article 29 Data Protection Working Party (2014). *Opinion 05/2014 on Anonymisation Techniques*. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf

Auger, A., & Doerr, B. (Eds.) (2011). *Theory of randomized search heuristics: Foundations and recent developments*. Singapore: World Scientific. https://doi.org/10.1142/7438

Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, *86*, 43–59. https://doi.org/10.1016/j.cie.2014.11.010

Bäck, T. (1996). *Evolutionary algorithms in theory and practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York: Oxford University Press. https://doi.org/10.1093/oso/9780195099713.001.0001

Bäck, T., Fogel, D. B., & Michalewicz, Z. (Eds.) (1997). *Handbook of evolutionary computation*. IOP Publishing/Oxford University Press. https://doi.org/10.1201/9780367802486

Bahner, J. E., Hüper, A.-D., & Manzey, D. (2008). Misuse of automated decision aids: Complacency, automation bias and the impact of training experience. *International Journal of Human-Computer Studies*, *66*(9), 688–699. https://doi.org/10.1016/j.ijhcs.2008.06.001

Baker, R. S., & Inventado, P. S. (2014). Educational Data Mining and Learning Analytics. In J. Larusson, & B. White (Eds.), *Learning Analytics* (pp. 61–75). Springer, New York, NY. https://doi.org/10.1007/978-1-4614-3305-7_4

Bangert, P. (2012). *Optimization for industrial problems*. Springer. https://doi.org/10.1007/978-3-642-24974-7

Bansal, M., Goyal, A., & Choudhary, A. (2022). A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. *Decision Analytics Journal*, *3*, 100071. https://doi.org/10.1016/j.dajour.2022.100071

Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). *Genetic programming: An introduction*. San Francisco (CA): Morgan Kaufmann. https://shop.elsevier.com/books/genetic-programming/banzhaf/978-0-08-050725-5

Bashab, A., Ibrahim, A. O., Tarigo Hashem, I. A., Aggarwal, K., Mukhlif, F., Ghaleb, F. A., & Abdelmaboud, A. (2023). Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues. *Computers, Materials & Continua*, *74*(3), 6461–6484. https://doi.org/10.32604/cmc.2023.034051

Baumann, B., & Kost, B. (1999). Random Cost method versus evolutionary algorithms. *Computational Optimization and Applications*, *14*(2), 203–218. https://doi.org/10.1023/A:1008795215794.

Baylari, A., & Montazer, G. A. (2009). Design a personalized e-learning system based on Item Response Theory and artificial neural network approach. *Expert Systems with Applications*, *36*(4), 8013–8021. https://doi.org/10.1016/j.eswa.2008.10.080.

Beasley, D., Bull, D. R., & Martin, R. R. (1993). An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, *15*(2), 56–69. https://orca.cardiff.ac.uk/id/eprint/64436/1/ga_%20overview1.pdf

Bellini, C., De Santis, A., Sannicandro, K., & Minerva, T. (2019). Data Management in Learning Analytics: terms and perspectives. *Journal of e-Learning and Knowledge Society*, *15*(3), 133-144. https://doi.org/10.20368/1971-8829/1135021

Benítez-Hidalgo, A., Nebro, A. J., García-Nieto, J., Oregi, I., & Del Ser, J. (2019). jMetalPy: A Python framework for multi-objective optimization with metaheuristics. *Swarm and Evolutionary Computation*, *51*, 100598. https://doi.org/10.1016/j.swevo.2019.100598

Beyer, H.-G., & Schwefel, H.-P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, *1*(1), 3–52. https://doi.org/10.1023/A:1015059928466

Bhattacharjee, D., Paul, A., Kim, J. H., & Karthigaikumar, P. (2018). An immersive learning model using evolutionary learning. *Computers & Electrical Engineering*, *65*, 236–249. https://doi.org/10.1016/j.compeleceng.2017.08.023

Biscani, F., & Izzo, D. (2020). A parallel global multiobjective framework for optimization: pagmo. *Journal of Open Source Software*, *5*(53), 2338. https://doi.org/10.21105/joss.02338.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer, https://link.springer.com/book/9780387310732

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical Secure Aggregation for Privacy-Preserving Machine Learning. *In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1175–1191. https://doi.org/10.1145/3133956.3133982

Bongard, J. C., Zykov, V., & Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, *314*(5802), 1118–1121. https://doi.org/10.1126/science.1133687

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier, & G. Saporta (Eds.), *Proceedings of COMPSTAT'2010* (pp. 177–186). Physica-Verlag. https://doi.org/10.1007/978-3-7908-2604-3_16

Boyer, S., & Veeramachaneni, K. (2015). Transfer learning for predictive models in massive open online courses. In C. Conati, N. Heffernan, A. Mitrovic, & M. Verdejo (Eds.), *Artificial Intelligence in Education (AIED 2015), LNCS 9112* (pp. 54–63). Springer. https://doi.org/10.1007/978-3-319-19773-9_6

Brochu, E., de Freitas, N., & Ghosh, A. (2007). Active preference learning with discrete choice data. In *Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS'07)* (pp. 409–416). Curran Associates Inc., Red Hook, NY, USA. https://dl.acm.org/doi/10.5555/2981562.2981614

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, I. L. McClelland, & B. Weerdmeester (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis. https://doi.org/10.1201/9781498710411-35.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Sys-*

*tems 33 (NeurIPS 2020)* (pp. 1877–1901). Curran Associates, Inc. Long version https://arxiv.org/abs/2005.14165

Burke, E. K., & Kendall, G. (2014). Introduction. In E. K. Burke, & G. Kendall (Eds.), *Search methodologies* (pp. 1–17). Springer. https://doi.org/10.1007/978-1-4614-6940-7_1

Camp, C. V., Pezeshk, S., & Cao, G. (1998). Optimized design of two-dimensional structures using a genetic algorithm. *Journal of Structural Engineering*, *124*(5), 551–559. https://doi.org/10.1061/(ASCE)0733-9445(1998)124:5(551)

Cantú-Paz, E. (2001). *Efficient and accurate parallel genetic algorithms*. Springer. https://doi.org/10.1007/978-1-4615-4369-5

CAST (2024). *Universal Design for Learning Guidelines version 3.0*. https://udlguidelines.cast.org

Ceschia, S., Di Gaspero, L., & Schaerf, A. (2023). Educational timetabling: Problems, benchmarks, and state-of-the-art results. *European Journal of Operational Research*, *308*, 1–18. https://doi.org/10.1016/j.ejor.2022.07.011

Chakraborty, B. (2014). Rough fuzzy consistency measure with evolutionary algorithm for attribute reduction. In *2014 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 723–728). IEEE. https://doi.org/10.1109/SMC.2014.6973995

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Chang, C. C. W., Ding, T. J., Ee, C. C. W., Han, W., Paw, J. K. S., Salam, I., Bhuiyan, M. A., & Kuan, G. S. (2024). Nature-inspired heuristic frameworks: Trends in solving multi-objective engineering optimization problems. *Archives of Computational Methods in Engineering*, *31*, 3551–3584. https://doi.org/10.1007/s11831-024-10090-x.

Chang, T.-Y., & Ke, Y.-R. (2013). A personalized e-course composition based on a genetic algorithm with forcing legality in an adaptive learning system. *Journal of Network and Computer Applications*, *36*(1), 533–542. https://doi.org/10.1016/j.jnca.2012.04.002

Charitopoulos, A., Rangoussi, M., & Koulouriotis, D. (2020).On the Use of Soft Computing Methods in Educational Data Mining and Learning Analytics Research: a Review of Years 2010–2018. *International Journal of Artificial Intelligence in Education*, *30*(3), 371–430. https://doi.org/10.1007/s40593-020-00200-8

Chen, C.-M. (2009). Ontology-based concept map for planning a personalised learning path. *British Journal of Educational Technology*, *40*(6), 1028–1058. https://doi.org/10.1111/j.1467-8535.2008.00892.x

Cheong, C. Y., Tan, K. C., & Veeravalli, B. (2007). A multiobjective evolutionary algorithm for examination timetabling. *Journal of Scheduling*, *10*(2), 121-146. https://doi.org/10.1007/s10951-008-0085-5

Christudas, B. C. L., Kirubakaran, E., & Thangaiah, P. R. J. (2018). An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials. *Telematics and Informatics*, *35*, 520–533. https://doi.org/10.1016/j.tele.2017.02.004

Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-36797-2

D'Mello, S., & Graesser, A. (2012). Dynamics of affective states during complex learning. *Learning and Instruction*, *22*(2), 145–157. https://doi.org/10.1016/j.learninstruc.2011.10.001

Darwin, C. (1859). *On the origin of species*. John Murray. https://darwin-online.org.uk/content/frameset?itemID=F373&viewtype=text&pageseq=1

Daskalaki, S., Birbas, T., & Housos, E. (2004). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, *153*(1), 117–135. https://doi.org/10.1016/S0377-2217(03)00103-6

De Jong, K. A. (1975). *Analysis of the behavior of a class of genetic adaptive systems* (Doctoral dissertation). University of Michigan. https://hdl.handle.net/2027.42/4507

De Santis, A. (2022). *Analisi multivariata e Learning Analytics: Metodi e applicazioni*. Pearson. https://it.pearson.com/docenti/universita/pearson-statistica.html

De Santis, A., Bellini, C., Jana, M., Righetti, L., Sannicandro, K., Talha, M., & Minerva, T. (2024). Dashboards for admins, teachers, and students: tools for monitoring and learning. In *EDULEARN24 Proceedings* (pp. 4666-4672). IATED. https://doi.org/10.21125/edulearn.2024.1150

De Santis, A., & Minerva, T. (2026). An Overview of Genetic Algorithms in Educational Research from 2020 to 2024. In F. Marcelloni, K. Madani, N. van Stein, & J. Filipe (Eds.), *Computational Intelligence. IJCCI 2025. Communications in Computer and Information Science, 2828*. Springer, Cham. https://doi.org/10.1007/978-3-032-15635-8_26

De Santis, A., Sannicandro, K., Bellini, C., & Minerva, T. (2024). Trends in the use of Multivariate Analysis in Educational Research: a review of methods and applications in 2018-2022. *Journal of E-Learning and Knowledge Society*, *20*(1), 47-55. https://doi.org/10.20368/1971-8829/1135946

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley. https://www.wiley.com/en-us/Multi+Objective+Optimization+using+Evolutionary+Algorithms-p-9780471873396

Deb, K. (2011). Multi-objective optimisation using evolutionary algorithms: An introduction. In L. Wang, A. Ng, & K. Deb (Eds.), *Multi-objective evolutionary optimisation for product design and manufacturing* (pp. 1–17). Springer. https://doi.org/10.1007/978-0-85729-652-8_1

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Deng, M., Wang, J., Hsieh, C. P., Wang, Y., Guo, H., Shu, T., Song, M., Xing, E. & Hu, Z. (2022). RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 3369–3391). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.222

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286. https://doi.org/10.1613/jair.105

Dofadar, D. F., Hayat Khan, R., Hasan, S., Anam Taj, T., Shakil, A., & Majumdar, M. (2022). *A hybrid evolutionary approach to solve university course allocation problem*. arXiv. https://arxiv.org/abs/2212.02230

Drachsler, H., & Greller, W. (2016). Privacy and analytics: it's a DELICATE issue a checklist for trusted learning analytics. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge (LAK '16)* (pp. 89–98). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2883851.2883893

Droste, S., Jansen, T., & Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, *276*(1–2), 51–81. https://doi.org/10.1016/S0304-3975(01)00182-7

Dunke, F., & Nickel, S. (2023). A matheuristic for customized multi-level multi-criteria university timetabling. *Annals of Operations Research*, *328*, 1313–1348. https://doi.org/10.1007/s10479-023-05325-2

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, *9*(3-4), 211–407. https://doi.org/10.1561/0400000042

Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing* (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-44874-8

El Yessefi, A., Cherrat, L., Ech-Charrat, M. R., & Ezziyyani, M. (2024). Implementing Genetic Algorithm-Based Expert Systems to Enhance Academic Career Planning. *International Journal of Information and Education Technology*, *14*(10), 1443-1452. https://doi.org/10.18178/ijiet.2024.14.10.2175

Elhaddad, Y. R., & Sallabi, O. (2011). A novel approach for combining genetic and simulated annealing algorithms. In S.-I. Ao, & L. Gelman (Eds.), *Electrical engineering and applied computing* (pp. 285–296). Springer. https://doi.org/10.1007/978-94-007-1192-1_24

Elshani, L., & Pireva Nuçi, K. (2021). *Constructing a personalized learning path using genetic algorithms approach*. arXiv. https://doi.org/10.48550/arXiv.2104.11276

Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, *20*(55), 1–21. http://jmlr.org/papers/v20/18-598.html

Escalante, H. J., Montes, M., & Sucar, L. E. (2015). ECBDL'14: The Evolutionary Computation for Big Data and Big Learning competition. *Knowledge-Based Systems*, *87*, 104–122. https://doi.org/10.1016/j.knosys.2015.05.030.

European Data Protection Board (2024). *Report of the work undertaken by the ChatGPT Taskforce*. https://www.edpb.europa.eu/system/files/2024-05/edpb_20240523_report_chatgpt_taskforce_en.pdf

European Union (2016). *Regulation (EU) 2016/679 (GDPR)*. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02016R0679-20160504

European Union (2024). *Regulation (EU) 2024/1689 (Artificial Intelligence Act)*. https://eur-lex.europa.eu/eli/reg/2024/1689/oj

Fakhrusy, M. R., & Widyani, Y. (2017). Moodle plugins for quiz generation using genetic algorithm. In *2017 International Conference on Data and Software Engineering (ICoDSE)*. IEEE.

Farsi, M., Hosahalli, D., Manjunatha, B. R., Gad, I., Atlam, E. S., Ahmed, A., Elmarhoumy, G., Elmarhoumy, M., & Ghoneim, O. A. (2021). Parallel genetic algorithms for optimizing the SARIMA model for better forecasting of the NCDC weather data. *Alexandria Engineering Journal*, *60*(1), 1299–1316. https://doi.org/10.1016/j.aej.2020.10.052

Ferguson, R. (2012). Learning analytics: Drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, *4*(5-6), 304–317. https://doi.org/10.1504/IJTEL.2012.051816

Fernández, A., García, S., Luengo, J., Bernadó-Mansilla, E., & Herrera, F. (2010). Genetics-based machine learning for rule induction: state of the art, taxonomy, and comparative study. *IEEE Transactions on Evolutionary Computation*, *14*(6), 913-941. doi:10.1109/TEVC.2009.2039140

Fernando, C., Banarse, D., Michalewski, H., Osindero, S., & Rocktaschel, T. (2023). *Prompt-Breeder: Self-referential self-improvement via prompt evolution*. arXiv. https://arxiv.org/abs/2309.16797

Floreano, D., Dürr, P., & Mattiussi, C. (2008). Neuroevolution: From architectures to learning. *Evolutionary Intelligence*, *1*, 47–62. https://doi.org/10.1007/s12065-007-0002-4

Fogel, D. B. (1995). *Evolutionary computation: Toward a new philosophy of machine intelligence*. IEEE Press.

Fogel, D. B. (2006). *Evolutionary computation: Toward a new philosophy of machine intelligence* (3rd ed.). Wiley-IEEE Press.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. Wiley.

Forrest, S., & Mitchell, M. (2016). Adaptive computation: The multidisciplinary legacy of John H. Holland. *Communications of the ACM*, *59*(8), 58–63. https://doi.org/10.1145/2964342

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, *13*, 2171–2175. https://dl.acm.org/doi/10.5555/2503308.2503311

Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 1322–1333). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2810103.2813677

Freitas, A. A. (2014). Comprehensible classification models: A position paper. *SIGKDD Explorations Newsletter*, *15*(1), 1–10. https://doi.org/10.1145/2594473.2594475

Galafassi, C., Galafassi, F. F. P., Vicari, R. M., & Reategui, E. B. (2023). EvoLogic: Toward an ITS for teaching propositional logic. *International Journal of Artificial Intelligence in Education*, *33*(1), 35-58. https://doi.org/10.1007/s40593-021-00287-7

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, *46*(4), 44:1–37. https://doi.org/10.1145/2523813

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, *64*(12), 86–92. https://doi.org/10.1145/3458723

Gen, M., & Cheng, R. (1997). *Genetic algorithms & engineering design*. Wiley. https://doi.org/10.1002/9780470172254

Glover, F., & Kochenberger, G. (Eds.). (2003). *Handbook of metaheuristics* (1st ed.). Springer. https://doi.org/10.1007/b101874

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org/

Guo, Q., Wang, R., Guo, J., Li, B., Song, K., Tan, X., Liu, G., Bian, J., & Yang, Y. (2024). Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *International Conference on Learning Representations (ICLR 2024)*. https://openreview.net/forum?id=ZG3RaNIsO8

Hafsa, M., Wattebled, P., Jacques, J., & Jourdan, L. (2025). Solving a multiobjective professional timetabling problem using evolutionary algorithms at Mandarine Academy. *International Transactions in Operational Research*, *32*(1), 244-269. https://doi.org/10.1111/itor.13276

Hajela, P., & Lin, C.-Y. (1992). Genetic search strategies in multi-criterion optimal design. *Structural Optimization*, *4*(2–3), 99–107. https://doi.org/10.1007/BF01759923

Halko, N., Martinsson, P.-G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, *53*(2), 217–288. https://doi.org/10.1137/090771806

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations*, *11*(1), 10–18. https://doi.org/10.1145/1656274.1656278

Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, *9*(2), 159–195. https://doi.org/10.1162/106365601750190398

Hao, J., Gan, J., & Zhu, L. (2022). MOOC performance prediction and personal performance improvement via Bayesian network. *Education and Information Technologies*, *27*, 7303–7326. https://doi.org/10.1007/s10639-022-10926-8

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)* (pp. 3315–3323). Curran Associates Inc., Red Hook, NY, USA. https://dl.acm.org/doi/10.5555/3157382.3157469

Harp, S. A., Samad, T., & Guha, A. (1989). Designing application-specific neural networks using the genetic algorithm. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 2* (pp. 447–454). Morgan Kaufmann. https://proceedings.neurips.cc/paper_files/paper/1989/hash/8c19f571e251e61cb8dd3612f26d5ecf-Abstract.html

Hart, S. G. (2006). NASA-Task Load Index (NASA-TLX); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(9), 904–908. https://doi.org/10.1177/154193120605000909

Hart, W.E., Smith, J.E., & Krasnogor, N. (2005). *Recent Advances in Memetic Algorithms*. Springer.

Hasan, R., Palaniappan, S., Mahmood, S., Abbas, A., Sarker, K. U., & Sattar, M. U. (2020). Predicting student performance in higher educational institutions using video learning analytics and data-mining techniques. *Applied Sciences*, *10*(11), 3894. https://doi.org/10.3390/app10113894

He, T. (2022). A Multiobjective Allocation Method for High-Quality Higher Education Resources Based on Cellular Genetic Algorithm. *Mathematical Problems in Engineering*, *2022*(1), 4322078. https://doi.org/10.1155/2022/4322078

Hlosta, M., Herrmannova, D., Vachova, L., Kuzilek, J., Zdrahal, Z., & Wolff, A. (2018). Modelling student online behaviour in a virtual learning environment. In *Proceedings of the 2014 Workshop on Learning Analytics and Machine Learning at the 2014 International Conference on Learning Analytics and Knowledge (LAK 2014)*. https://arxiv.org/abs/1811.06369

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The University of Michigan Press.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (Revised ed.). MIT Press. https://doi.org/10.7551/mitpress/1090.001.0001

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2019). *The curious case of neural text degeneration*. arXiv:1904.09751. https://arxiv.org/abs/1904.09751

Hou, X., & Luo, M. (2022). Research on artificial intelligence search algorithm for optimization problem. In *Proceedings of SPIE 12456, International Conference on Artificial Intelligence and Intelligent Information Processing (AIIIP 2022)*, 124560S. SPIE. https://doi.org/10.1117/12.2660066

Hssina, B., & Erritali, M. (2019). A personalized pedagogical objectives based on a genetic algorithm in an adaptive learning system. *Procedia Computer Science*, *151*, 1152–1157. https://doi.org/10.1016/j.procs.2019.04.164

Huang, M.-J., Huang, H.-S., & Chen, M.-Y. (2007). Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Systems with Applications*, *33*(3), 551–564. https://doi.org/10.1016/j.eswa.2006.05.019.

Hurmuzi, A., Kubaji, A., Hamdoon, W., Makki, Z., Azeez, M., & Denizhan, B. (2023). Course timetabling for distance learning program at SKU using genetic algorithm. In *2023 Second International Conference on Advanced Computer Applications (ACA)* (pp. 1-7). IEEE. https://doi.org/10.1109/ACA57612.2023.10346857

IEEE Standards Association. (2023, October 6). *IEEE Standard for Learning Technology-JavaScript Object Notation (JSON) data model format and Representational State Transfer (RESTful) web service for learner experience data tracking and access (xAPI) (IEEE 9274.1.1-2023)*. https://standards.ieee.org/ieee/9274.1.1/7321/

Ifenthaler, D., & Yau, J. Y. K. (2020). Utilising learning analytics to support study success in higher education: A systematic review. *Educational Technology Research and Development*, *68*, 1961–1990. https://doi.org/10.1007/s11423-020-09788-z

Information Commissioner's Office (ICO). (n.d.). *What is automated decision-making and profiling?* https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/individual-rights/automated-decision-making-and-profiling/what-is-automated-individual-decision-making-and-profiling/ (retrieved on August 25th, 2025)

ISO/IEC. (2022). *ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection-Information security management systems-Requirements*. https://www.iso.org/standard/82875.html (retrieved on August 25th, 2025)

Izzo, D. (2012). PyGMO and PyKEP: Open-source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization). *5th International Conference on Astrodynamics Tools and Techniques (ICATT)*.

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C. & Kavukcuoglu, K. (2017). *Population Based Training of neural networks*. arXiv preprint. https://arxiv.org/abs/1711.09846

Jin, Y. (2005). A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, *9*, 3–12. https://doi.org/10.1007/s00500-003-0328-5

Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, *1*(2), 61–70. https://doi.org/10.1016/j.swevo.2011.05.001

Jivet, I., Scheffel, M., Specht, M., & Drachsler, H. (2018, March). License to evaluate: Preparing learning analytics dashboards for educational practice. In *Proceedings of the 8th international conference on learning analytics and knowledge* (pp. 31-40). https://doi.org/10.1145/3170358.3170421

Jiwatode, M., Schlecht, L., & Dockhorn, A. (2024). Online optimization of curriculum learning schedules. *2024 IEEE Conference on Games (CoG)*, 1–8. https://doi.org/10.1109/CoG60054.2024.10645570

Kairouz, P., McMahan, H. B., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, *14*(1–2), 1–210. https://doi.org/10.1561/2200000083

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., StAdvance Distributed Learninger, M., Weller, J., Kuhn, J., & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, *103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE. https://doi.org/10.1109/ICNN.1995.488968

Khare, P. (2015). Learning analytics of postgraduate sociology students of Indira Gandhi National Open University, India. *Learning Community – An International Journal of Educational and Social Development*, *6*(2–3), 231–247. https://doi.org/10.5958/2231-458X.2015.00022.6

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

Kinshuk, Chen, NS., Cheng, IL., Chew, S.W. (2016). Evolution is not enough: Revolutionizing Current Learning Environments to Smart Learning Environments. *International Journal of Artificial Intelligence in Education*, 26, 561–581 (2016). https://doi.org/10.1007/s40593-016-0108-x

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Kizilcec, R. F., Saltarelli, A. J., Reich, J., & Cohen, G. L. (2017). Closing global achievement gaps in MOOCs. Brief interventions address social identity threat at scale. *Science*, *355*(6322), 251–252. https://doi.org/10.1126/science.aap9550

Kovacs, T. (2012). Genetics-Based Machine Learning. In G. Rozenberg, T. Bäck, & J.N. Kok (Eds.), *Handbook of Natural Computing*. Springer: Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-92910-9_30

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*. MIT Press. https://mitpress.mit.edu/9780262111706

Kučak, D., Juričić, V., & Đambić, G. (2019). Application of genetic algorithms in higher education area. In *Proceedings of the 30th DAAAM International Symposium* (pp. 343–347). https://doi.org/10.2507/30th.daaam.proceedings.045

Kuzilek, J., Hlosta, M., & Zdrahal, Z. (2017). Open University Learning Analytics dataset. *Scientific Data*, *4*, 170171. https://doi.org/10.1038/sdata.2017.171

Kwakye, B. D., Li, Y., Mohamed, H. H., Baidoo, E., & Asenso, T. Q. (2024). Particle guided metaheuristic algorithm for global optimization and feature selection problems. *Expert Systems with Applications*, *248*, 123362. https://doi.org/10.1016/j.eswa.2024.123362

Lang, C., Siemens, G., Wise, A. F., Gašević, D., & Merceron, A. (Eds.) (2022). *The Handbook of Learning Analytics* (2nd ed.). Society for Learning Analytics Research. https://solaresearch.org/wp-content/uploads/hla22/HLA22.pdf

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10(2), 141–160. https://doi.org/10.1016/j.cogsys.2006.07.004

Lei, Y., Shi, J., & Yan, Z. (2018). A memetic algorithm based on MOEA/D for the examination timetabling problem. *Soft Computing*, *22*(5), 1511-1523. https://doi.org/10.1007/s00500-017-2886-y

León, N.S., & Castillo Domínguez, P.G. (2020). Genetic Algorithm in the Allocation of Hours of an Institute of Higher Education. *18th LACCEI International Multi-Conference for Engineering, Education, and Technology, 29-31 July 2020, Buenos Aires, Argentina*. http://dx.doi.org/10.18687/LACCEI2020.1.1.518

Li, X., Ouyang, F., & Chen, W. (2022). Examining the effect of a genetic-algorithm-enabled grouping method on collaborative performances, processes, and perceptions. *Journal of Computing in Higher Education*, *34*, 790–819. https://doi.org/10.1007/s12528-022-09321-6

Liu, S., Chen, C., Qu, X., Tang, K., & Ong, Y.-S. (2024). Large Language Models as Evolutionary Optimizers. In *IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–8). IEEE. https://doi.org/10.1109/CEC58653.2024.10393232

Llorà, X., Sastry, K., Alías, F., Goldberg, D. E., & Welge, M. (2006). Analyzing active interactive genetic algorithms using visual analytics. In *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation* (pp. 1417–1418). https://doi.org/10.1145/1143997.1144223

Lockett, A. J. (2020). Optimization in finite spaces. In A. J. Lockett (Ed.), *General-purpose optimization through information maximization* (pp. 43–65). Springer. https://doi.org/10.1007/978-3-662-62007-6_3

Long, P., & Siemens, G. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*. https://er.educause.edu/articles/2011/9/penetrating-the-fog-analytics-in-learning-and-education

Luke, S. (2013). *Essentials of metaheuristics* (2nd ed.). Lulu. https://cs.gmu.edu/~sean/book/metaheuristics/

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 30. https://papers.nips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Luo, Y., & Niu, P. (2024). Physical education teaching scheduling technology based on chaotic genetic algorithm. *Scientific Reports*, *14*(1), 27912. https://doi.org/10.1038/s41598-024-79646-y

Lv, L., & Zhao, J. (2018). The firefly algorithm with Gaussian disturbance and local search. *Journal of Signal Processing Systems*, *90*(8–9), 1123–1131. https://doi.org/10.1007/s11265-017-1278-y

Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology*, *106*(4), 901–918. https://doi.org/10.1037/a0037123

Mahlous, A. R., & Mahlous, H. (2023). Student timetabling genetic algorithm accounting for student preferences. *PeerJ Computer Science*, *9*, e1200. https://doi.org/10.7717/peerj-cs.1200

Márquez-Vera, C., Cano, A., Romero, C., Noaman, A. Y. M., Fardoun, H. M., & Ventura, S. (2016). Early dropout prediction using data mining: A case study with high school students. *Expert Systems*, *33*(1), 107–124. https://doi.org/10.1111/exsy.12135

Masiello, I., Mohseni, Z., Palma, F., Nordmark, S., Augustsson, H., & Rundquist, R. (2024). A current overview of the use of learning analytics dashboards. *Education Sciences*, *14*(1), 82. https://doi.org/10.3390/educsci14010082

Masters, D., & DeLaurentis, D. (2001, January). Aerodynamic shape optimization using genetic algorithms (AIAA-2001-0251). In *39th AIAA Aerospace Sciences Meeting and Exhibit*. https://doi.org/10.2514/6.2001-0251

MathWorks (2025). *Global Optimization Toolbox*. https://www.mathworks.com/products/global-optimization.html

McKenney, S., & Reeves, T. C. (2021). Educational design research: Portraying, conducting, and enhancing productive scholarship. *Medical education*, *55*(1), 82-92. https://doi.org/10.1111/medu.14280

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)* (pp. 1273–1282). PMLR. https://proceedings.mlr.press/v54/mcmahan17a.html

Mebane Jr., W. R., & Sekhon, J. S. (2011). Genetic optimization using derivatives: The rgenoud package for R. *Journal of Statistical Software*, *42*(11), 1–26. https://doi.org/10.18637/jss.v042.i11

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, *54*(6), Article 115. https://doi.org/10.1145/3457607 (open version: https://arxiv.org/abs/1908.09635)

Merz, P., & Freisleben, B. (1999). A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)* (pp. 2063–2070). IEEE. https://doi.org/10.1109/CEC.1999.785529

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs* (3nd ed.). Springer. https://doi.org/10.1007/978-3-662-03315-9

Michalewicz, Z., & Fogel, D. B. (2004). *How to solve it: Modern heuristics* (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-07807-5

Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzyan, A., Duffy, N., & Hodjat, B. (2017). *Evolving deep neural networks* (arXiv:1703.00548). https://doi.org/10.48550/arXiv.1703.00548

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267, 1–38. https://doi.org/10.1016/j.artint.2018.07.007

Minerva, T., De Santis, A., Sannicandro, K., & Bellini, C. (2024a). Integration of Educational Clusters with Open Badges and Blended Intensive Program (ECOBI): a comprehensive approach to future university education. *Journal of E-Learning and Knowledge Society*, 20(3), 112-125. https://doi.org/10.20368/1971-8829/1136068

Minerva, T., De Santis, A., Bellini, C., & Sannicandro, K. (2024b). Students in Italian online universities: enrollments time series analysis from 2005 to 2021. *Journal of E-Learning and Knowledge Society*, 19(4), 50-75. https://doi.org/10.20368/1971-8829/1135872

Minerva, T., & Poli, I. (2001a). A neural net model to predict high tides in Venice. In S. Borra, R. Rocci, M. Vichi, & M. Schader (Eds.), *Advances in classification and data analysis* (pp. 367–373). Springer. https://doi.org/10.1007/978-3-642-59471-7_45

Minerva, T., & Poli, I. (2001b). Building ARMA models with genetic algorithms. In E. J. W. Boers (Ed.), *Applications of Evolutionary Computing. EvoWorkshops 2001, LNCS 2037* (pp. 335–342). Springer. https://doi.org/10.1007/3-540-45365-2_35

Minerva, T., Paterlini, S., & Poli, I. (2000). GANND: A genetic algorithm for predictive neural network design-A financial application. *Economics & Complexity*, *2*, 4–14.

Mir, S.M., Khan, N.A. A systematic review of MOOC platforms, evaluation frameworks and development of integrated model. *Discover Education*, *5*, 105 (2026). https://doi.org/10.1007/s44217-025-01085-2

Mirjalili, S., & Dong, J. S. (2020). Introduction to multi-objective optimization. In S. Mirjalili & J. S. Dong, *Multi-objective optimization using artificial intelligence techniques* (pp. 1–9). Springer. https://doi.org/10.1007/978-3-030-24835-2_1

Mitchell, M. (1995). Genetic algorithms: An overview. *Complexity*, *1*(1), 31–39. https://doi.org/10.1002/cplx.6130010108

Mitchell, M. (1996). *An introduction to genetic algorithms*. MIT Press.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., & Gebru, T. (2019). Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19)* (pp. 220–229). Association for Computing Machinery, New York. https://doi.org/10.1145/3287560.3287596

Molnar, C. (2019). *Interpretable machine learning: A guide for making black box models explainable*. Leanpub. https://christophm.github.io/interpretable-ml-book/

Moreno, J., Ovalle, D. A., & Vicari, R. M. (2012). A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics. *Computers & Education*, *58*(1), 560–569. https://doi.org/10.1016/j.compedu.2011.09.011

Moriarty, D. E., Schultz, A. C., & Grefenstette, J. J. (1999). Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, *11*, 241–276. https://doi.org/10.1613/jair.613

Moscato, P. (1989). *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms (Caltech Concurrent Computation Program, C3P Report)*. Caltech Library.

Muklason, A., Parkes, A. J., Özcan, E., McCollum, B., & McMullan, P. (2017). Fairness in examination timetabling: Student preferences and extended formulations. *Applied Soft Computing*, *55*, 302–318. https://doi.org/10.1016/j.asoc.2017.01.026

Müller, T. (2016). Real-life examination timetabling. *Journal of Scheduling*, *19*(3), 257–270. https://doi.org/10.1007/s10951-014-0391-z

Müller, T., & Rudová, H. (2016). Real-life curriculum-based timetabling with elective courses and course sections. *Annals of Operations Research*, *239*, 153–170. https://doi.org/10.1007/s10479-014-1643-1

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press. https://mitpress.mit.edu/9780262018029/

Narayanan, A., & Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. *2008 IEEE Symposium on Security and Privacy* (pp. 111–125). https://doi.org/10.1109/SP.2008.33

National Institute of Standards and Technology (NIST) (2020). *NIST Special Publication 800-207: Zero Trust Architecture*. https://doi.org/10.6028/NIST.SP.800-207

National Institute of Standards and Technology (NIST) (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1*. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

Neumann, F., & Witt, C. (2010). *Bioinspired computation in combinatorial optimization: Algorithms and their computational complexity*. Springer. https://doi.org/10.3991/ijet.v16i11.21017

Ngo, S. T., Jaafar, J. B., Aziz, I. A., Nguyen, G. H., & Bui, A. N. (2021). Genetic Algorithm for Solving Multi-Objective Optimization in Examination Timetabling Problem. *International Journal of Emerging Technologies in Learning (iJET)*, *16*(11), pp. 4–24. https://doi.org/10.3991/ijet.v16i11.21017

Nguyen, A., Gardner, L., & Sheridan, D. (2020). Data analytics in higher education: An integrated view. *Journal of Information Systems Education*, *31*(1), 61–71. https://jise.org/Volume31/n1/JISEv31n1p61.html

Nguyen, T. T., Yang, S., & Branke, J. (2012). Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, *6*, 1-24. https://doi.org/10.1016/j.swevo.2012.05.001

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-40065-5

Nogueira, J. V. d. C. M. F., Bernardino, H. S., Souza, J. F. d., Gonçalves, L. B., & Soares, S. S. R. F. (2024). Exploring the solution space for adaptive curriculum sequencing: Study of a multi-objective approach. *Internet of Things*, *25*, 101052. https://doi.org/10.1016/j.iot.2023.101052

Nolfi, S., & Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT Press.

Obayashi, S., & Takanashi, S. (1996). Genetic optimization of target pressure distributions for inverse design methods. *AIAA Journal*, *34*(5), 881–886. https://doi.org/10.2514/3.13163

Olson, R. S., Bartley, N., Urbanowicz, R. J., & Moore, J. H. (2016). Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of GECCO '16* (pp. 485–492). https://doi.org/10.1145/2908812.2908918

OpenAI (2023). *GPT-4 technical report*. arXiv. https://arxiv.org/abs/2303.08774v6

Ouyang, F., Zheng, L., & Jiao, P. (2022). Artificial intelligence in online higher education. A systematic review of empirical research from 2011 to 2020. *Education and Information Technologies*, *27*, 7893–7925. https://doi.org/10.1007/s10639-022-10925-9

Papamitsiou, Z., & Economides, A. A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Educational Technology & Society*, *17*(4), 49–64. https://www.jstor.org/stable/jeductechsoci.17.4.49

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, *39*(2), 230–253. https://doi.org/10.1518/001872097778543886

Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. (2002). Data mining with an ant colony algorithm. *IEEE Transactions on Evolutionary Computation*, *6*(4), 321–332. https://doi.org/10.1109/TEVC.2002.802452

Paterlini, S., & Minerva, T. (2010a). Regression model selection using genetic algorithms. In V. Munteanu (Ed.), *Recent Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing* (pp. 19–27). WSEAS Press.

Paterlini, S., & Minerva, T. (2010b). Genetic algorithms in partitional clustering: A comparison. In V. Munteanu (Ed.), *Recent advances in neural networks, fuzzy systems & evolutionary computing* (pp. 28–36). WSEAS Press.

Pattarin, F., Paterlini, S., & Minerva, T. (2004). Clustering financial time series: An application to mutual funds style analysis. *Computational Statistics & Data Analysis*, *47*(2), 353–372. https://doi.org/10.1016/j.csda.2003.11.009

Pitzer, E., & Affenzeller, M. (2012). A comprehensive survey on fitness landscape analysis. In J. Fodor, R. Klempous, & C. P. Suárez Araujo (Eds.), *Recent advances in intelligent engineering systems* (Studies in Computational Intelligence, Vol. 378, pp. 161–191). Springer. https://doi.org/10.1007/978-3-642-23229-9_8

Plante, J.-F., Larocque, M., & Adès, M. (2024). Objective model selection with parallel genetic algorithms using an eradication strategy. *The Canadian Journal of Statistics*, *52*(2), 636–654. https://doi.org/10.1002/cjs.11775

Potter, M. A., & De Jong, K. A. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, *8*(1), 1–29. https://doi.org/10.1162/106365600568086

Price, K. V., Storn, R. M., & Lampinen, J. A. (2005). *Differential evolution: A practical approach to global optimization*. Springer. https://doi.org/10.1007/3-540-31306-0

Puchinger, J., & Raidl, G. R. (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In J. Mira, & J. R., Álvarez (Eds.), *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach.*

*IWINAC 2005*. Lecture Notes in Computer Science, vol 3562. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11499305_5

Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., & O'Sullivan, J. M. (2022). A review of feature selection methods for machine learning-based disease risk prediction. *Frontiers in bioinformatics*, *2*, 927312. https://doi.org/10.3389/fbinf.2022.927312

Qiu, M. (2022). Balanced Allocation of Educational Resources Based on Parallel Genetic Algorithm. *Mathematical Problems in Engineering*, 2022(1), 7517267. https://doi.org/10.1155/2022/7517267

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. G., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12(1), 55–89. https://doi.org/10.1007/s10951-008-0077-5

Queiroga, E. M., Lopes, J. L., Kappel, K., Aguiar, M., Araújo, R. M., Munoz, R., Villarroel, R., & Cechinel, C. (2020). A learning analytics approach to identify students at risk of dropout. A Case Study with a Technical Distance Education Course. *Applied Sciences*, *10*(11), 3998. https://doi.org/10.3390/app10113998

Rajeev, S., & Krishnamoorthy, C. S. (1992). Discrete optimization of structures using genetic algorithms. *Journal of Structural Engineering*, *118*(5), 1233–1250. https://doi.org/10.1061/(ASCE)0733-9445(1992)118:5(1233)

Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020). Analyzing and predicting students' performance by means of machine learning: A review. *Applied Sciences*, *10*(3), 1042. https://doi.org/10.3390/app10031042

Real, E., Aggarwal, A., Huang, Y., & Le, Q. V. (2019). Regularized evolution for image classifier architecture search. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(1), 4780–4789. https://doi.org/10.1609/aaai.v33i01.33014780

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog.

Reich, J. (2022). Learning analytics and learning at scale. In C. Lang, G. Siemens, A. F. Wise, D. Gašević, & A. Merceron (Eds.), *Handbook of learning analytics* (2nd ed., pp. 188–195). Society for Learning Analytics Research (SoLAR).

Reich, J., & Ruipérez-Valiente, J. A. (2019). The MOOC pivot. *Science*, *363*(6423), 130–131. https://doi.org/10.1126/science.aav7958

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP-IJCNLP 2019* (pp. 3982–3992). https://doi.org/10.18653/v1/D19-1410

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). https://doi.org/10.1145/2939672.2939778

Ridley, M. (2006). *Genome: The autobiography of a species in 23 chapters*. Harper Perennial.

Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (applications and reviews)*, *40*(6), 601-618.

Ronchi, D., Tosca, E. M., Bartolucci, R., & Magni, P. (2024). Go beyond the limits of the genetic algorithm in daily covariate selection practice. *Journal of Pharmacokinetics and Pharmacodynamics*, *51*, 109–121. https://doi.org/10.1007/s10928-023-09875-7

Rosin, C. D., & Belew, R. K. (1997). New methods for competitive coevolution. *Evolutionary Computation*, *5*(1), 1–29. https://doi.org/10.1162/evco.1997.5.1.1

Rudolph, G. (1997). Convergence properties of evolutionary algorithms. Verlag Dr. Kovac. (https://www.researchgate.net/publication/220689559_Convergence_properties_of_evolutionary_algorithms)

Ruipérez-Valiente, J. A., Staubitz, T., Jenner, M., Halawa, S., Zhang, J., Despujol, I., Maldonado-Mahauad, J., Montoro, G., Peffer, M., Rohloff, T., Lane, J., Turro, C., Li, X., Pérez-Sanagustín, M., & Reich, J. (2022). Large scale analytics of global and regional MOOC providers: Differences in learners' demographics, preferences, and perceptions. *Computers & Education*, *180*, 104426. https://doi.org/10.1016/j.compedu.2021.104426

Sahlaoui, H., Hessane, A., Agoujil, S., Tekouabou, S. C. K., & Barebzi, A. (2024). Enhancing student performance in African smart cities: a web-based approach through advanced ensemble modeling and genetic feature optimization. *Data & Policy*, *6*, e33. https://doi.org/10.1017/dap.2024.25

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). *Toolformer: Language models can teach themselves to use tools*. arXiv:2302.04761. https://arxiv.org/abs/2302.04761

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM*, *63*(12), 54–63. https://doi.org/10.1145/3381831

Schwefel, H.-P. (1995). *Evolution and optimum seeking*. Wiley.

Schwendimann, B. A., Rodriguez-Triana, M. J., Vozniuk, A., Prieto, L. P., Boroujeni, M. S., Holzer, A., Gillet, D., & Dillenbourg, P. (2017). Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies*, *10*(1), 30–41. https://doi.org/10.1109/TLT.2016.2599522

Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software*, *53*(4), 1–37. https://doi.org/10.18637/jss.v053.i04

Scrucca, L. (2017). On some extensions to GA package: Hybrid optimisation, parallelisation and islands evolution. *The R Journal*, *9*(1), 187–206. https://doi.org/10.32614/RJ-2017-008

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28*. https://papers.nips.cc/paper_files/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

Sedrakyan, G., Mannens, E., & Verbert, K. (2019). Guiding the choice of learning dashboard visualizations: Linking dashboard design and data visualization concepts. *Journal of Computer Languages*, *50*, 19-38. https://doi.org/10.1016/j.jvlc.2018.11.002

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin Company.

Shih, T. K., Gunarathne, W. K. T. M., Ochirbat, A., & Su, H. M. (2018). Grouping peers based on complementary degree and social relationship using genetic algorithm. *ACM Transactions on Internet Technology (TOIT)*, *19*(1).

Shin-ike, K., & Iima, H. (2008). An inspection of a system for improving learning abilities by using a neural network and a genetic algorithm. In *2008 34th Annual Conference of IEEE Industrial Electronics, Orlando, FL, USA, 2008* (pp. 3479-3484). https://doi.org/10.1109/IECON.2008.4758521

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy* (pp. 3–18). https://doi.org/10.1109/SP.2017.41

Siemens, G., & Baker, R.S. (2012). Learning analytics and educational data mining: towards communication and collaboration. *LAK'12: Proceedings of the 2nd International Confer-*

*ence on Learning Analytics and Knowledge*, 252-254. https://doi.org/10.1145/2330601.2330661

Siew, E. S. K., Goh, S. L., Kendall, G., Sabar, N. R., & Abdullah, S. (2024). A survey of solution methodologies for exam timetabling problems. *IEEE Access*, *12*, 41479-41498. https://doi.org/10.1109/ACCESS.2024.3378054

Slade, S., & Prinsloo, P. (2013). Learning analytics: Ethical issues and dilemmas. *American Behavioral Scientist*, *57*(10), 1510–1529. https://doi.org/10.1177/0002764213479366

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, NIPS 2012* (pp. 2951-2959). (Advances in Neural Information Processing Systems; Vol. 4) (https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf)

Sreenath, K., & Jeyakumar, G. (2016). Evolutionary algorithm-based rule(s) generation for personalized courseware construction in educational data mining. In *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, India, 2016* (pp. 1–7). https://doi.org/10.1109/ICCIC.2016.7919631

Stanley, K. O., D'Ambrosio, D. B., & Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, *15*(2), 185–212. https://doi.org/10.1162/artl.2009.15.2.15202

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, *10*(2), 99–127. https://doi.org/10.1162/106365602320169811

Storn, R., & Price, K. (1997). Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

Stracke, C.M., Downes, S., Conole, G., Burgos, D., Nascimbeni, F. (2019). Are MOOCs open educational resources? A literature review on history, definitions and typologies of OER and MOOCs. *Open Praxis*, *11*(4), 331–341. https://doi.org/10.5944/openpraxis.11.4.1010

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645–3650). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1355

Suganuma, M., Shirakawa, S., & Nagao, T. (2017). A genetic programming approach to designing convolutional neural network architectures. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17)* (pp. 497–504). https://doi.org/10.1145/3071178.3071229

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(5), 557–570. https://doi.org/10.1142/S0218488502001648

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, *89*(9), 1275–1296. https://doi.org/10.1109/5.949485

Tan, C. J., Lim, T. Y., Bong, C. W., & Liew, T. K. (2017). A multi-objective evolutionary algorithm-based soft computing model for educational data mining: A distance learning experience. *Asian Association of Open Universities Journal*, *12*(1), 106–123. https://doi.org/10.1108/AAOUJ-01-2017-0012

Tan, X., Jia, Y., Liu, W., & Hu, C. (2012). Personalized course generation and evolution based on genetic algorithms. *Journal of Zhejiang University-SCIENCE C*, *13*(4), 287–298. https://doi.org/10.1631/jzus.C1200174

Tran, B., Xue, B., & Zhang, M. (2019). Genetic programming for multiple-feature construction on high-dimensional classification. *Pattern Recognition*, *93*, 404–417. https://doi.org/10.1016/j.patcog.2019.05.006

Ugarte, N., Aranzabal, A., Arruarte, A., & Larrañaga, M. (2022). Using the behavioural tendency of students in a team environment for team formation. In *2022 IEEE Frontiers in Education Conference (FIE)*. IEEE.

Ungerer, L., & Slade, S. (2022). Ethical considerations of artificial intelligence in learning analytics in distance education contexts. In P. Prinsloo P., Slade, S., & Khalil, M. (Eds.), *Learning Analytics in Open and Distributed Learning* (pp. 105–120). Springer. https://doi.org/10.1007/978-981-19-0786-9_8

Ustun, B., Spangher, A., & Liu, Y. (2019). Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19)* (pp. 10–19). https://doi.org/10.1145/3287560.3287566

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197–221. https://doi.org/10.1080/00461520.2011.611369

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, 30. (Open access version: https://arxiv.org/abs/1706.03762)

Verbert, K., Duval, E., Klerkx, J., Govaerts, S., & Santos, J. L. (2013). Learning analytics dashboard applications. *American Behavioral Scientist*, *57*(10), 1500-1509. https://doi.org/10.1177/0002764213479363

Vlahavas, I., & Vrakas, D. (2008). *Artificial intelligence for advanced problem solving techniques*. IGI Global. https://doi.org/10.4018/978-1-59904-705-8

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, *31*(2), 841–887. https://doi.org/10.2139/ssrn.3063289

Waheed, H., Hassan, S.-U., Aljohani, N. R., Hardman, J., Alelyani, S., & Nawaz, R. (2020). Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior*, *104*, 106189. https://doi.org/10.1016/j.chb.2019.106189

Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F., & Wilson, J. (2020). The What-If Tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, *26*(1), 56–65. https://doi.org/10.1109/TVCG.2019.2934619

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, *4*(2), 65–85. https://doi.org/10.1007/BF00175354

Woolf, B. P. (2009). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.

Xia, X., & Xie, Z. (2001). DAMBE: Software package for data analysis in molecular biology and evolution. *Journal of Heredity*, *92*(4), 371–373. https://doi.org/10.1093/jhered/92.4.371

Xie, S.-T., He, Z.-B., Chen, Q., Chen, R.-X., Kong, Q.-Z., & Song, C.-Y. (2021). Predicting learning behavior using log data in blended teaching. *Scientific Programming*, *2021*, 4327896. https://doi.org/10.1155/2021/4327896

Xing, W., Guo, R., Petaković, E., & Goggins, S. (2015). Participation-based student final performance prediction model through interpretable genetic programming: Integrating learning analytics, educational data mining and theory. *Computers in Human Behavior*, *47*, 168–181. https://doi.org/10.1016/j.chb.2014.09.034

Xu, Y., Zhang, G., Yan, C., Wang, G., Jiang, Y., & Zhao, K. (2021). A two-stage multi-objective optimization method for envelope and energy generation systems of primary and secondary school teaching buildings in China. *Building and Environment*, *204*, 108142. https://doi.org/10.1016/j.buildenv.2021.108142

Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606–626. https://doi.org/10.1109/TEVC.2015.2504420

Yalçınkaya, A., Gedik Balay, İ., & Şenoğlu, B. (2021). A new approach using the genetic algorithm for parameter estimation in multiple linear regression with long-tailed symmetrically distributed error terms: An application to the COVID-19 data. *Chemometrics and Intelligent Laboratory Systems*, *216*, 104372. https://doi.org/10.1016/j.chemolab.2021.104372

Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., & Chen, X. (2024). *Large language models as optimizers (OPRO)*. Published as a conference paper at ICLR 2024. https://arxiv.org/abs/2309.03409

Yang, X.-S. (2010a). *Nature-inspired metaheuristic algorithms* (2nd ed.). Luniver Press.

Yang, X.-S. (2010b). A new metaheuristic bat-inspired algorithm. In J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, & N. Krasnogor (Eds.), *Nature inspired cooperative strategies for optimization (NICSO 2010)* (Studies in Computational Intelligence, Vol. 284, pp. 65–74). Springer. https://doi.org/10.1007/978-3-642-12538-6_6

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). *ReAct: Synergizing reasoning and acting in language models*. arXiv:2210.03629. https://arxiv.org/abs/2210.03629.

Ying, C., Klein, A., Real, E., Christiansen, E., Murphy, K., & Hutter, F. (2019). NAS-Bench-101: Towards reproducible neural architecture search. *Proceedings of the 36th International Conference on Machine Learning*. https://proceedings.mlr.press/v97/

Yokota, T., Gen, M., & Li, Y.-X. (1996). Genetic algorithm for non-linear mixed integer programming problems. *Computers & Industrial Engineering*, *30*(4), 905–917. https://doi.org/10.1016/0360-8352(96)00041-1

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., & Gummadi, K. P. (2017). Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 1171–1180). https://doi.org/10.1145/3038912.3052660

Zakaria, A. L. J., Anas, B. O., & Oucamah, C. M. M. (2022). Intelligent system for personalised interventions and early drop-out prediction in MOOCs. *International Journal of Advanced Computer Science and Applications*, *13*(9), 700-710. https://dx.doi.org/10.14569/IJACSA.2022.0130983

Zandavi, S. M., Chung, V., & Anaissi, A. (2021). Multi-user remote lab: Timetable scheduling using simplex nondominated sorting genetic algorithm. *ACM/IMS Transactions on Data Science*, *2*(2), 1-13. https://doi.org/10.1145/3437260

Zhang, Q. (2022). An optimized solution to the course scheduling problem in universities. *Journal of Intelligent Systems*, *31*(1), 1065-1073. https://doi.org/10.1515/jisys-2022-0114

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731. https://doi.org/10.1109/TEVC.2007.892759

Zhang, R., Meng, Z., Wang, H., Liu, T., Wang, G., Zheng, L., & Wang, C. (2024). Hyperscale data analysis oriented optimization mechanisms for higher education management systems platforms with evolutionary intelligence. *Applied Soft Computing*, *155*, 111460. https://doi.org/10.1016/j.asoc.2024.111460

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, *1*(1), 32–49. https://doi.org/10.1016/j.swevo.2011.03.001

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2023). *Large language models are human-level prompt engineers*. Published as a conference paper at ICLR 2023. https://arxiv.org/abs/2211.01910

Zitzler, E., Laumanns, M., & Thiele, L. (2001). *SPEA2: Improving the strength Pareto evolutionary algorithm*. TIK Report 103. https://doi.org/10.3929/ethz-a-004284029

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
EDUNEXT Italian Universities Digital Education Hub

# GENETIC ALGORITHMS AND EVOLUTIONARY COMPUTATION IN EDUCATION

**Tommaso Minerva**, is a Full Professor of Statistics at the University of Modena and Reggio Emilia. Holding a Ph.D. in Solid State Physics, his early research focused on computational simulations of the electronic structure of superconducting oxides.
Subsequently, his interests shifted towards computational techniques for data analysis, and since the late 1990s, he has been involved in biology-inspired algorithms (genetic and evolutionary algorithms, artificial neural networks, fuzzy logic), which are precursors to modern artificial intelligence algorithms. More recently, his research interests have turned to the application of statistical-computational approaches in the field of education and learning analytics. Tommaso Minerva is the Scientific Coordinator of the EDUNEXT project.

**Annamaria De Santis**, is a Tenure-Track Researcher in Experimental Pedagogy at the University of Modena and Reggio Emilia. She holds a Ph.D. in Pedagogy and Educational Sciences and worked as an Instructional Designer at the Edunova Centre (2015-2023).
Managing Editor of the Journal of e-Learning and Knowledge Society, she authored the book "Analisi multivariata e learning analytics: metodi e applicazioni" (Multivariate Analysis and Learning Analytics: Methods and Applications). Her teaching and research interests focus on the application of quantitative approaches to educational research and on the design of digital educational programmes, environments, and solutions.
She directs the Master's in Digital Education Design and is part of the EDUNEXT coordination team.

Pearson

9 788891 945464