

I choose you...

Improving Genetic Algorithms by Association Rules

- Pokémon Trading Card Game: competitive game with an active tournament scene
- Building strong decks becomes increasingly difficult, especially for inexperienced players
- Deckbuilding is a combinatorial optimization problem: a deck must contain exactly 60 cards, but not every combination is useful
- Successful decks depend on synergies between Pokémon, Trainer, and Energy cards

Research Question

To what extent can association rules and genetic algorithms be combined to create a practical, data-driven, and tournament-oriented deckbuilder for the Pokémon Trading Card Game?

Core Idea

The proposed system combines two complementary techniques:

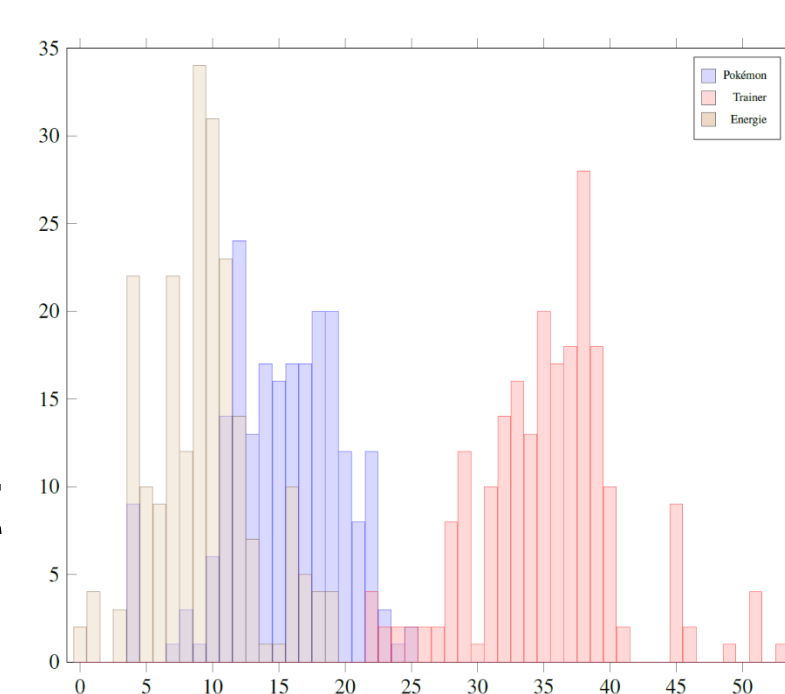
- **Association rules** capture patterns from tournament decks
- **Genetic algorithms** explore the large deckbuilding search space
- Guided optimization process towards meaningful card combinations

1) Data Sources

- **Pokémon TCG API**: General card pool, including metadata such as card type, set, and regulation mark
- **Limitless TCG**: Real tournament decklists as empirical reference data

3) Deck Structure

- Decks must contain exactly 60 cards
- Approximate target ranges based on reference decks:
 - Pokémon: 10–20 cards
 - Trainer: 30–40 cards
 - Energy: 4–14 cards
- Constraints reduce the search space and prevent unrealistic deck structures



2) Preprocessing

- Card pool is filtered to focus on the Standard format with regulation marks F, G, and H
- Cards are grouped into three categories: Pokémon, Trainer and Energy
- User specifies a Pokémon type
- Search space build around cards belonging to that type

4) Association Rules

- Mined from real tournament decklists
- Frequently co-occurring card combinations → Indicate strategic synergy
- Evaluation of relationships using:
 - **Support**: How often card combination appears
 - **Confidence**: How likely one card appears given another
 - **Lift**: How strong the relationship is compared to chance

5) Fitness Function

- Quality of generated decks is measured using a weighted combination of association rule metrics:

$$\text{fitness} = \text{support} + 1.2 \times \text{confidence} + 2 \times \text{lift}$$
- The fitness function turns random variation into directed optimization

Implementation

- Python 3.13.1
- Main libraries:
 - **pandas** 3.0.1 for card and deck data processing
 - **NumPy** 2.4.3 for numerical operations
 - **Matplotlib** 3.10.8 for visualizations
 - **mlxtend** 0.24.0 for association rules



6) Genetic Algorithm

- Follows an evolutionary process:
 - **Initial generation**: Candidate decks are created using extracted frequent patterns
 - **Mutation**: Cards are replaced, usually by cards of the same category, to preserve deck structure
 - **Crossover**: Parent decks are recombined to create new candidate decks
 - **Selection**: Decks are scored; the best candidates are selected for next generation
- Process continues until
 - Target fitness value is reached or
 - Maximum number of iterations is exceeded