

AI Techniques Overview

● **Evolutionary Computation** – Genetic Algorithms



● **Symbolic Reasoning** – Prolog

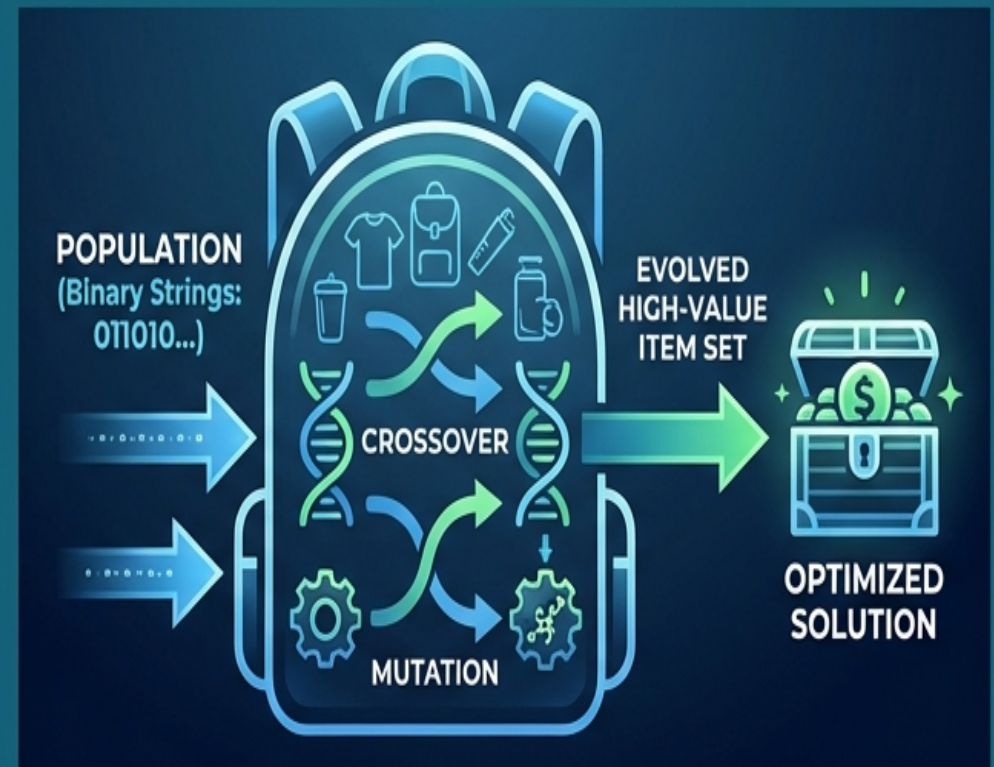


● **Combinatorial Search** – CSP



Genetic Algorithm for Knapsack

- 1. Randomly generate populations of item sets
- 2. Crossover & mutation evolve better solutions
- 3. Fitness = total value respecting weight limit



Prolog Course Knowledge Base

- Facts: courses, prerequisites, instructors
- Rules enable prerequisite & instructor queries
- Supports description, list, and check operations

```
% Facts
course(csc101, "Intro to CS").
course(csc201, "Data Structures").
course(csc301, "Algorithms").


prerequisite(csc201, csc101).
prerequisite(csc301, csc201).

instructor(csc101, "Dr. Smith").
instructor(csc201, "Prof. Johnson").
instructor(csc301, "Dr. Lee").

% Sample Rule
can_take(Student, Course) :-
    prerequisite(Course, Prereq),
    has_taken(Student, Prereq).
```


CSP Sports Scheduling

- Variables: match slots, venues, teams
- Constraints: no conflicts, fairness, venue availability
- Backtracking/forward-checking finds feasible schedule

	Slot 1	Slot 2	Slot 3	Slot 4
Team A				
Team B				
Team C				
Team D				

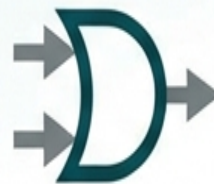
**Constraint Satisfaction**

Practical Takeaways

- AI methods map to real-world problems
- GA yields near-optimal knapsack solutions
- **Prolog** enables logical course queries
- CSP produces conflict-free sports schedules



Genetic Algorithms (GA)



Prolog Logic



Constraint Satisfaction (CSP)