Computational Intelligence

LMU Munich
winter term 2024/2025

Thomas Gabor
Claudia Linnhoff-Popien

finding policies...

Algorithm 1 (brute force (policy))**.** Let $\mathcal{A}$ be a set of actions. Let $\mathcal{O}$ be a set of observations. Let $\Gamma \subseteq (\mathcal{O} \to \mathcal{A}) \to \mathbb{B}$ be a space of goal predicates on policy functions. Let $\gamma \in \Gamma$ be a goal predicate. We assume that the policy space $\Pi \subseteq \mathcal{O} \to \mathcal{A}$ is enumerable, i.e., $\Pi = \langle \pi_i \rangle_{i \in \mathbb{N}}$. Brute force starting from i is given via the function

$$b(i) = \begin{cases} \pi_i & \text{if } \gamma(\pi_i), \\ b(i+1) & \text{otherwise.} \end{cases}$$

If not further specified, the call to $b(0)$ is called brute force search for an agent policy. Usually, an additional termination condition is specified.

finding policies...

Algorithm 2 (random search (policy)). Let $\mathcal{A}$ be a set of actions. Let $\mathcal{O}$ be a set of observations. Let $\Gamma \subseteq (\mathcal{O} \to \mathcal{A}) \to \mathbb{B}$ be a space of goal predicates on policy functions. Let $\gamma \in \Gamma$ be a goal predicate. We assume that the policy space $\Pi \subseteq \mathcal{O} \to \mathcal{A}$ can be sampled from, i.e., $\pi \sim \Pi$ returns a random element from Π. Random search for n samples is given via the function

$$\rho(n) = \begin{cases} \emptyset & \text{if } n = 0, \\ \pi & \text{if } n > 0 \text{ and } \gamma(\pi) \text{ where } \pi \sim \Pi, \\ \rho(n-1) & \text{otherwise.} \end{cases}$$

The Goal Class Hierarchy

Goal Class 5: State Values

Goal Class 4: Rewards and Costs

Goal Class 3: Goal Direction

Goal Class 2: Goal Valuation

Goal Class 1: Goal Predicate | Goal Class 1.5: Multiple Goal Predicates

Goal Class 0: No Goals

Goal Class 1.5: Multiple Goal Predicates

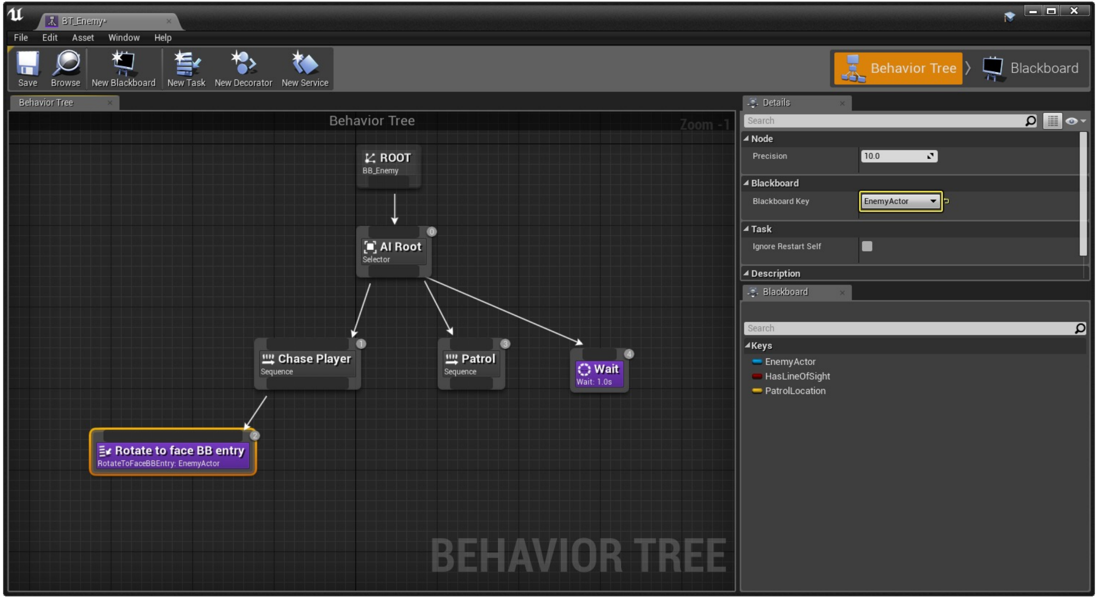
"I know it is good when I see it — with more aspects!"

encoding policies...

From Test to Test Suite

encoding policies...

Behavior Trees



https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/ArtificialIntelligence/BehaviorTrees/BehaviorTreeQuickStart/

Combinatorial Search

finding policies…

The Goal Class Hierarchy

Goal Class 5: State Values

Goal Class 4: Rewards and Costs

Goal Class 3: Goal Direction

Goal Class 2: Goal Valuation

Goal Class 1: Goal Predicate | Goal Class 1.5: Multiple Goal Predicates

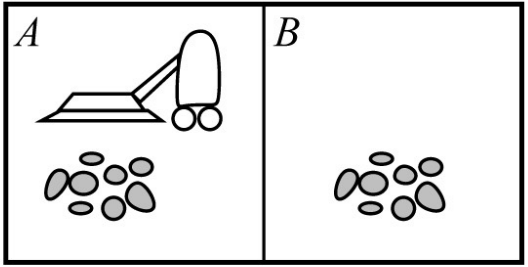
Goal Class 0: No Goals

Goal Class 2: Goal Valuation

"I know how good it is when I see it!"

running example #1

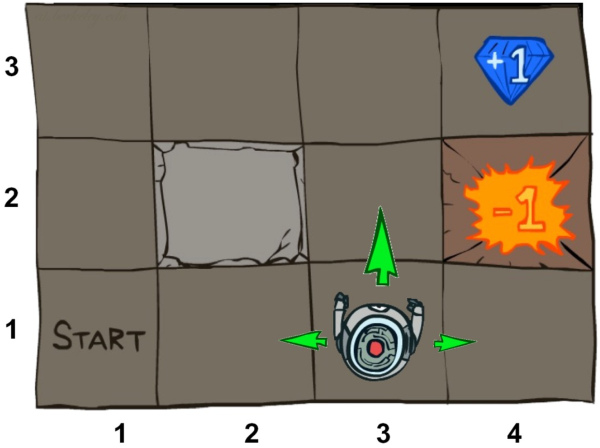
The Vacuum World



Russel, Norvig. Artificial Intelligence – A Modern Approach. Third Edition. 2016.

running example #2

The Basic Grid World



https://inst.eecs.berkeley.edu/~cs188/fa22/

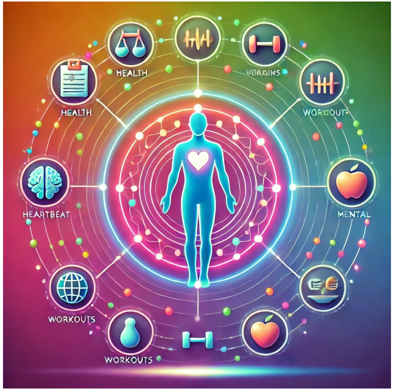
running example #3

Resource/Stock Trading

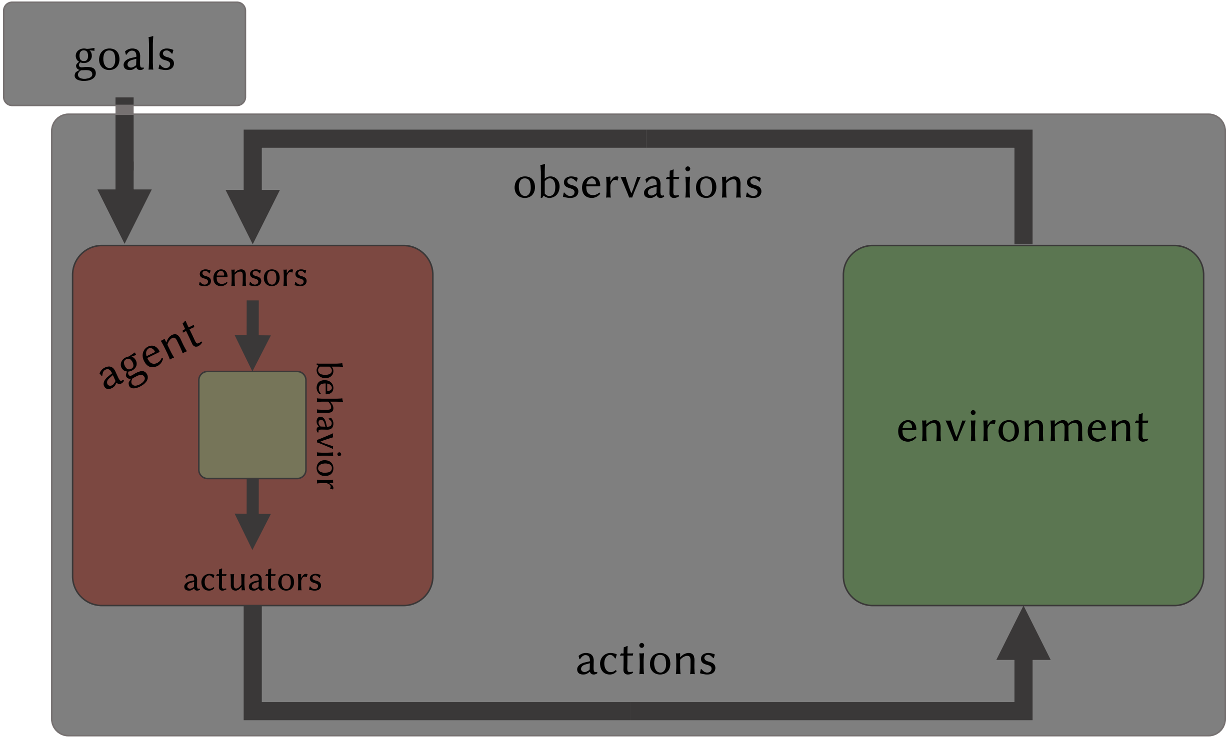
chatgpt.com

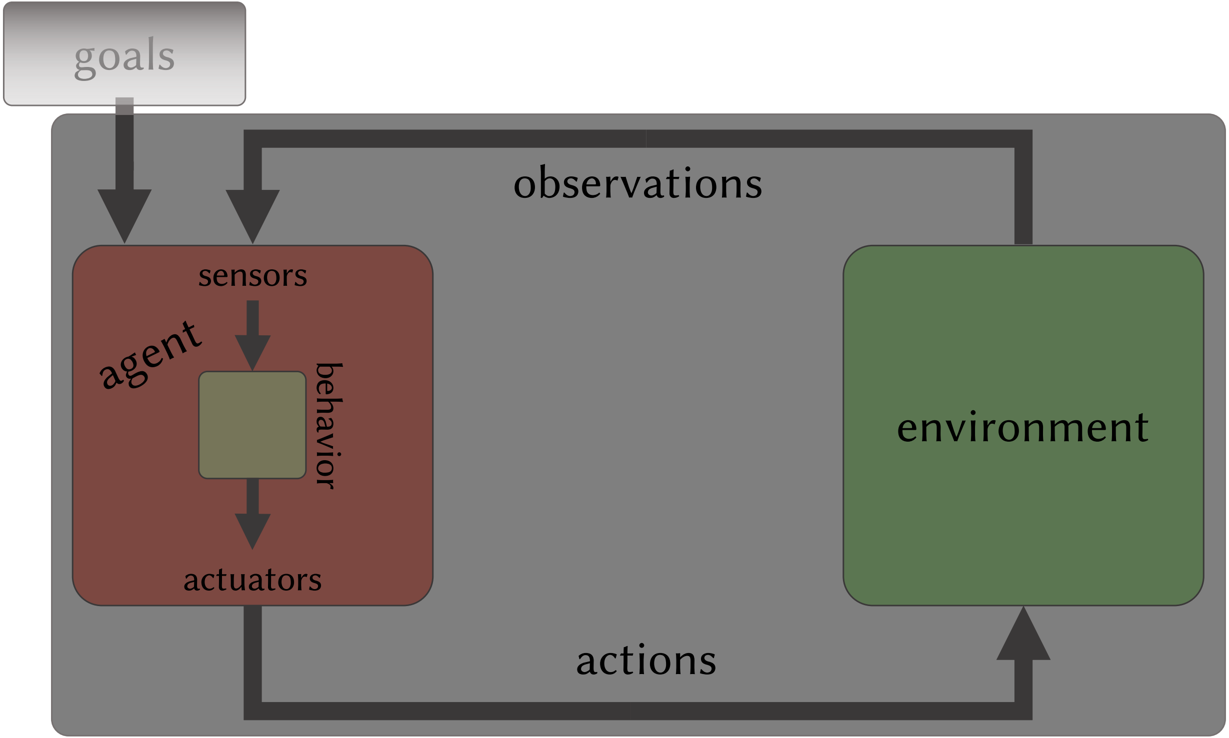
running example #4

Personal Life Assistant



chatgpt.com


goals
observations
sensors
agent
behavior
environment
actuators
actions


goals
observations
sensors
agent
behavior
actuators
environment
actions

finding policies...

Weighted Random Sampling

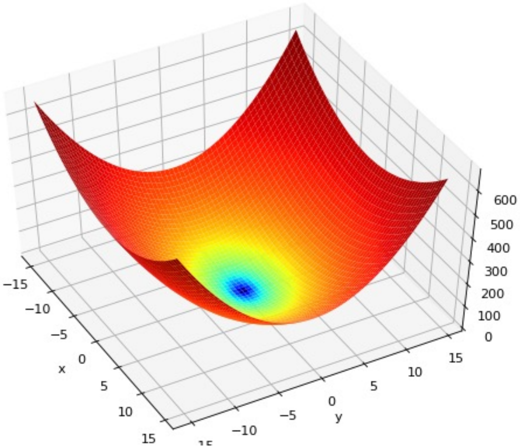
finding policies…

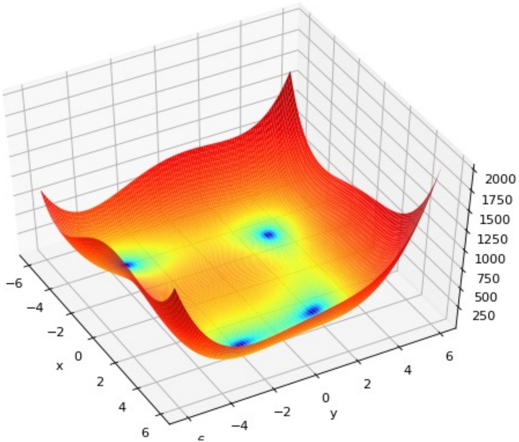
Definition 2 (optimization)**.** Let $\mathcal{X}$ be an arbitrary state space. Let $\mathcal{T}$ be an arbitrary set called target space and let $\leq$ be a total order on $\mathcal{T}$. A total function $\tau : \mathcal{X} \to \mathcal{T}$ is called target function. Optimization (minimization/maximization) is the procedure of searching for an $x \in \mathcal{X}$ so that $\tau(x)$ is optimal (minimal/maximal). Unless stated otherwise, we assume minimization. An optimization run of length $g + 1$ is a sequence of states $\langle x_t \rangle_{0 \leq t \leq g}$ with $x_t \in \mathcal{X}$ for all t.

Let $e : \langle \mathcal{X} \rangle \times (\mathcal{X} \to \mathcal{T}) \to \mathcal{X}$ be a possibly randomized or non-deterministic function so that the optimization run $\langle x_t \rangle_{0 \leq t \leq g}$ is produced by calling e repeatedly, i.e., $x_{t+1} = e(\langle x_u \rangle_{0 \leq u \leq t}, \tau)$ for all t, $1 \leq t \leq g$, where x_0 is given externally (e.g., $x_0 =_{def} 42$) or chosen randomly (e.g., $x_0 \sim \mathcal{X}$). An optimization process is a tuple $(\mathcal{X}, \mathcal{T}, \tau, e, \langle x_t \rangle_{0 \leq t \leq g})$.

Definition 3 (optimization (policy))**.** Let $\mathcal{X} = \Pi$ be a policy space. Let $\mathcal{D} = (\Pi, \mathcal{T}, \tau, e, \langle x_t \rangle_{0 \leq t \leq g})$ be an optimization process according to Definition 2. $\mathcal{D}$ is called a policy optimization process.

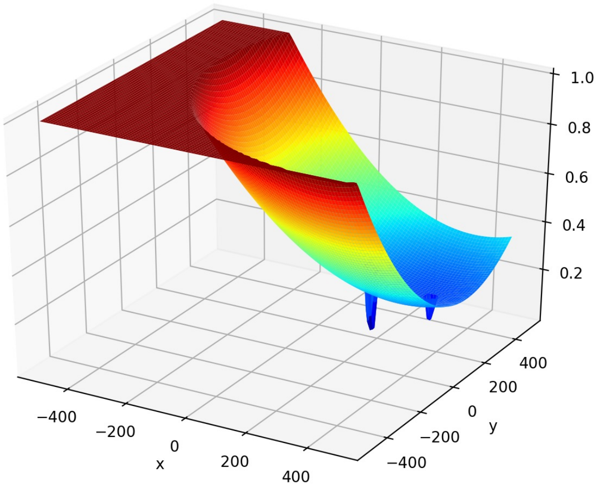
The Solution Landscape Metaphor

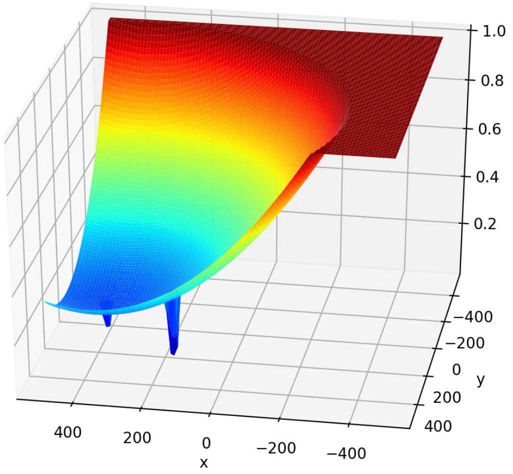




source: deap.readthedocs.io/en/master/api/benchmarks.html

The Solution Landscape Metaphor





The Solution Landscape Metaphor