Learning in Rational Agents

Stuart Russell
Computer Science Division
UC Berkeley

- - - - - - - - - - - - - - - - - - -

Joint work with Eric Wefald, Devika Subramanian, Shlomo Zilberstein, Gary Ogasawara, Ron Parr, John Binder, Keiji Kanazawa, Daphne Koller, Jonathan Tash, and Daishi Harada

Outline

1. Intelligence and Rationality
2. Rationality and Tetris
3. Tetris: A Modern Approach

Intelligence

Need *constructive, formal, broad* definitions—Int—relating *input/structure/output* and *Intelligence*

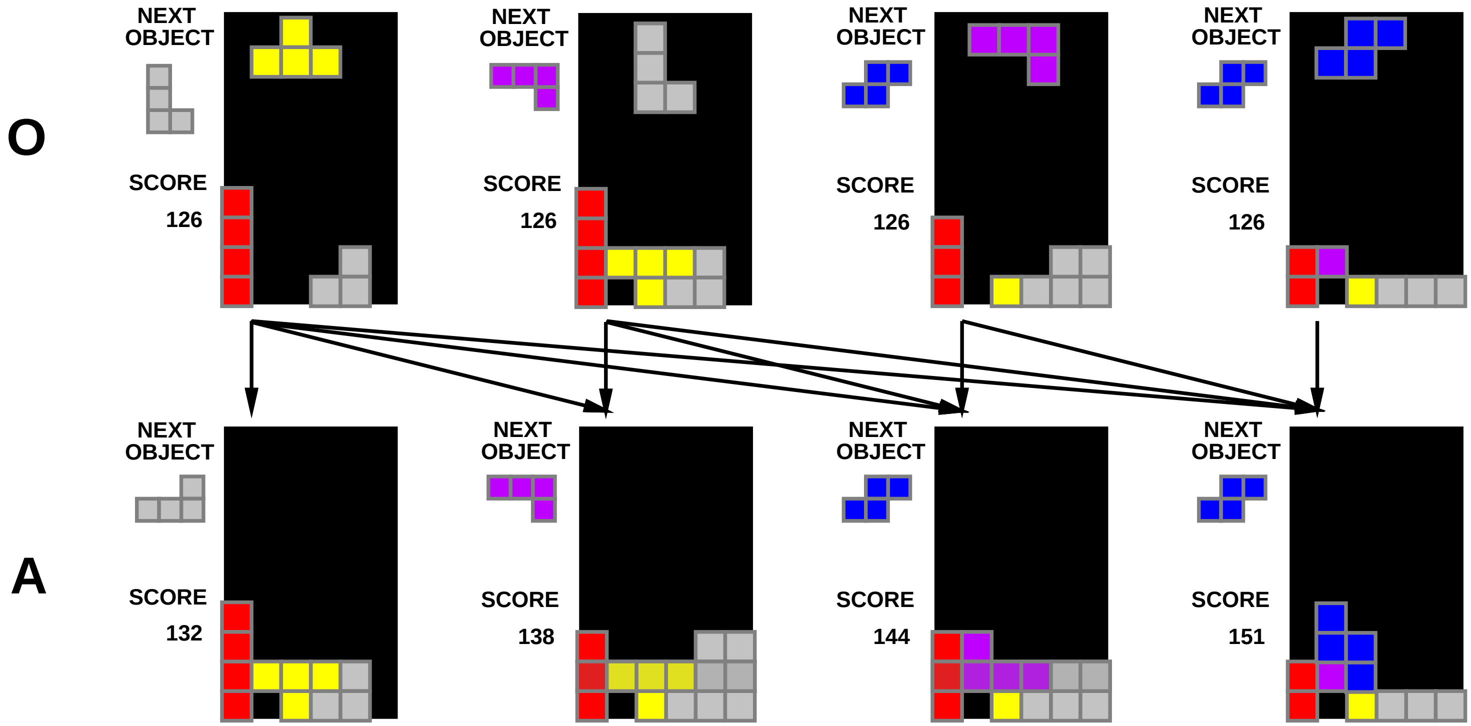
“Look! My system is Int!”

- Is the claim interesting?
- Is the claim sometimes true?
- What research do we do on Int?

Candidates formal definitions of *Intelligence* are:

- Int_1: Perfect rationality
- Int_2: Calculative rationality
- Int_3: Metalevel rationality
- Int_4: Bounded optimality

Agents and environments



Agents perceive **O** and act **A** in environment E

An *agent function* f : $\mathbf{O}^* \rightarrow \mathbf{A}$

specifies an act for any percept sequence

Global measure $V(f, E)$ evaluates f in E

Int_1 = perfect rationality

Agent f_{opt} is perfectly rational:

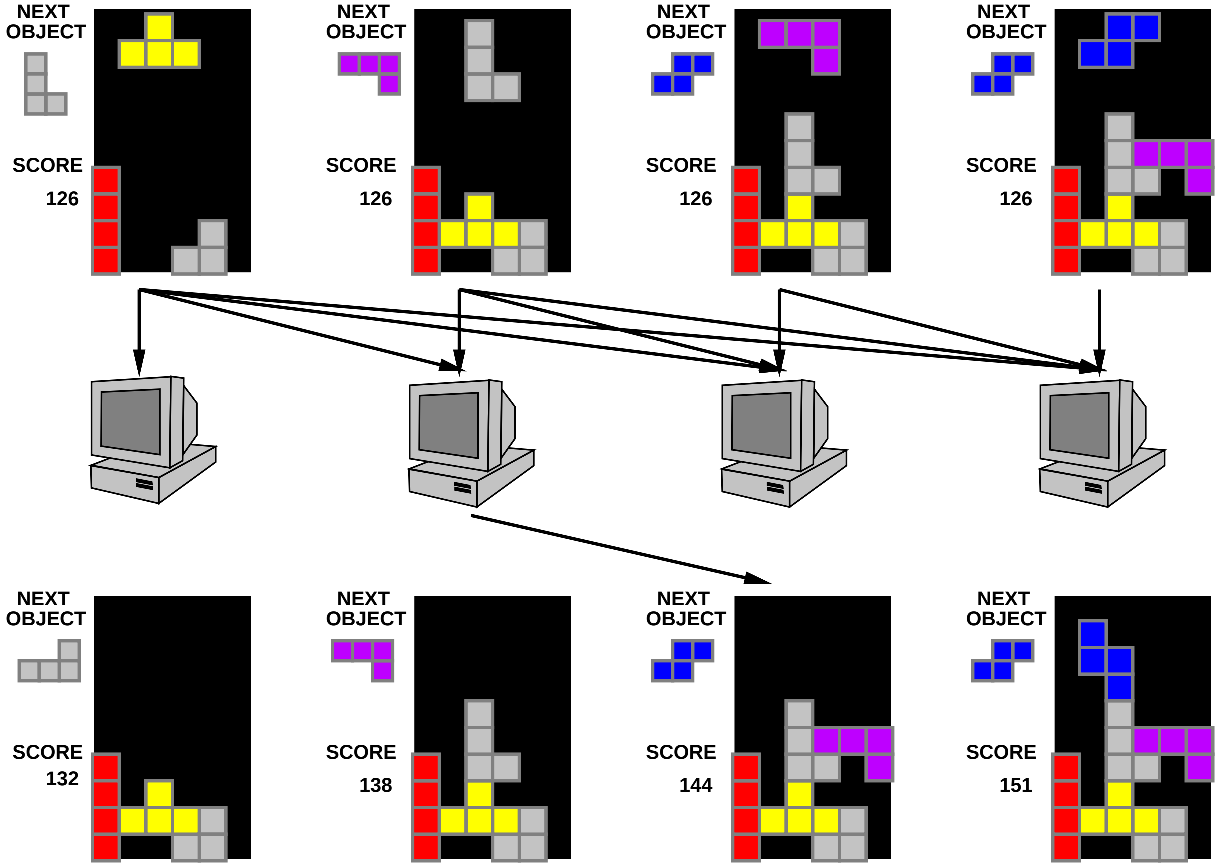
$$f_{opt} = argmax_f \; V(f, \mathbf{E})$$

i.e., the best possible behaviour

"Look! My system is perfectly rational!"

- Very interesting claim
- *VERY seldom possible*
- Research relates *global* measure to *local* constraints, e.g., maximizing utility

Machines and programs



Agent is a machine M running a program p

This defines an agent function $f = Agent(p, M)$

Int_2 = calculative rationality

p is calculatively rational if $Agent(p, M) = f_{opt}$

when M is infinitely fast

i.e., p eventually computes the best action

"Look! My system is calculatively rational!"

☹ *Useless in real-time* worlds*

☺ Quite often true

☺ Research on calculative tools, e.g.
logical planners, probabilistic networks

Int_3: metalevel rationality

$Agent(p, M)$ is metalevelly rational if it controls its computations optimally (I. J. Good's Type II)

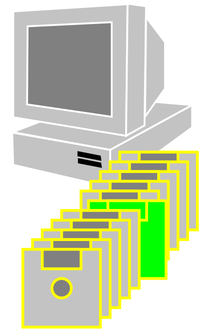
"Look! My system is metalevelly rational!"

😈 Very interesting claim

☹ *VERY seldom possible*

😈 Research on rational metareasoning

Int_4: bounded optimality



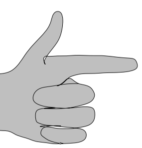
$Agent(p_{opt}, M)$ is bounded-optimal iff

$$p_{opt} = argmax_p V(Agent(p, M), \mathbf{E})$$

i.e., the best program given M.

Look! My system is bounded-optimal!

- Very interesting claim
- Always possible
- Research on all sorts of things



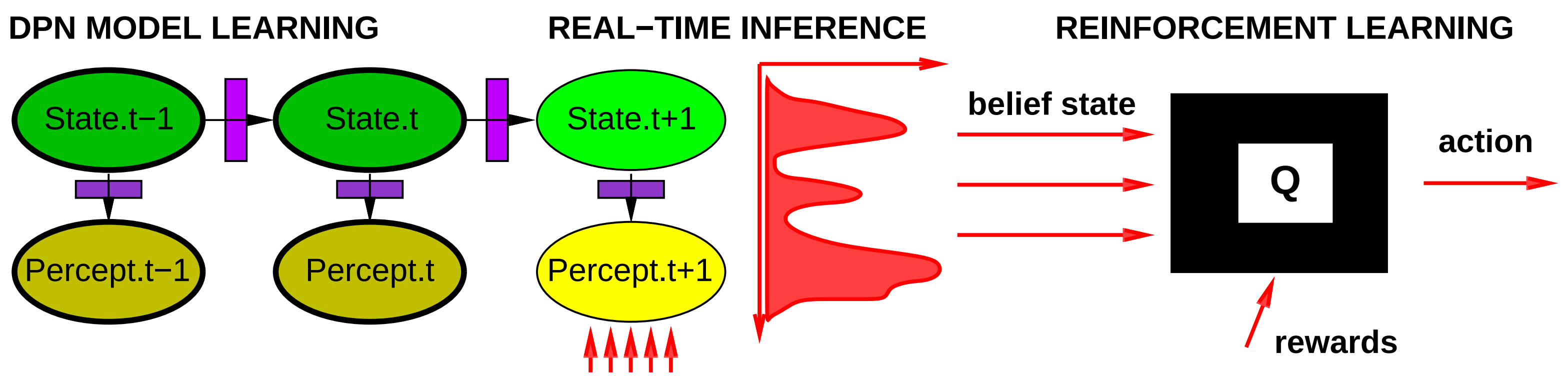
Bounded optimality can substitute for Intelligence

See also philosophy (Dennett), game theory (Wigderson, Papadimitriou)

Progress on calculative rationality

Real world: nondeterminism, partial observability $\Rightarrow$ POMDP

Correct decision determined by *belief state*



Combine the following:

- ♢ DPNs for belief state representation [OR, IJCAI 93]
- ♢ ER/SOF for efficient approximate inference [KKR, UAI 95]
- ♢ DPN learning for learning model [RBKK, IJCAI 95]
- ♢ Reinforcement learning to create utility model [PR, NIPS 97]

Progress on metalevel rationality

Do the Right Thinking:

- ♢ Computations are *actions*
- ♢ Cost=time   Benefit=better decisions
- ♢ Value $\approx$ benefit minus cost

General agent program:

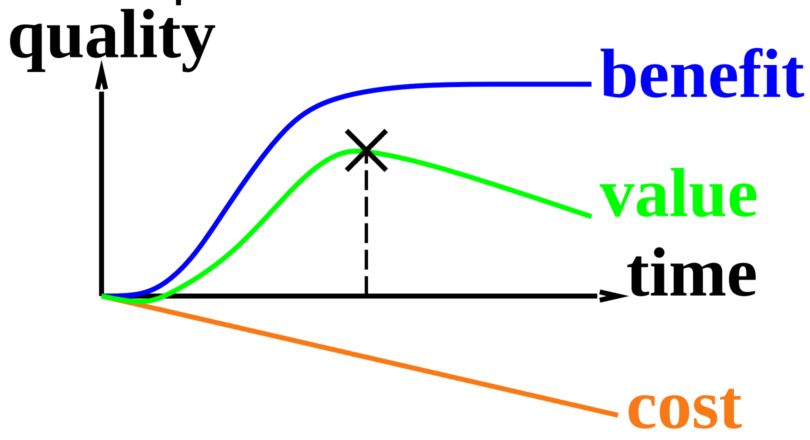
Repeat until no computation has value > 0:

Do the best computation

Do the current best action

Anytime algorithms

Decision quality that improves over time



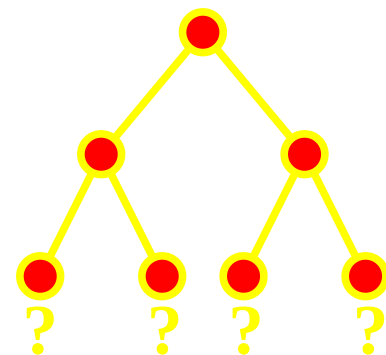
Rational metareasoning applies trivially

Anytime tools becoming a big industry

Fine-grained metareasoning

Explicit model of effects of computations

$\Rightarrow$ selection as well as termination



Compiled into efficient formula

for value of computation

Applications in search, games, MDPs show improvement over standard algorithms

Algorithms in AI

Metareasoning could/should replace devious algorithms

Research on bounded optimal agent design

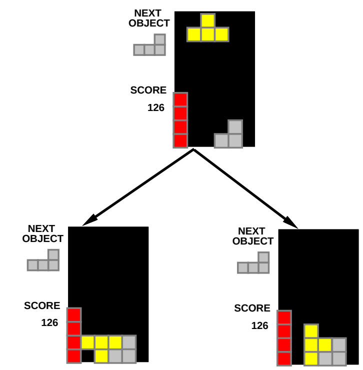
Bounded optimality imposes *nonlocal* constraints on action

$\Rightarrow$ Optimize over programs, not actions

Research agenda—still fairly conventional:

- ♢ Convergence to bounded optimality in simple designs
- ♢ Bounded optimality of metalevel systems
- ♢ Bounded optimality of composite systems
- ♢ Dominance among various agent structures
- ♢ Radical bounded optimality

Simple design I



Tetris agent

- ◇ Depth-1 search with value function V
- ◇ V has *fixed runtime* (e.g., NN)

No time limit

$\Longrightarrow$ standard RL converges* to BO agent

"God's eye view" problem handled automatically

What if the agent does a depth-2 search?

Time limit for whole game (e.g., chess)

Same "cycle time" but *different* BO agent

RL in SMDP converges* to BO agent [Harada, AAAI 97]

Feedback mechanism design is crucial

Simple design II

Tetris agent

- ♢ Depth-1 search with value function V
- ♢ V is a *variable-runtime* function approximator
  accuracy varies with runtime (e.g., decision tree)

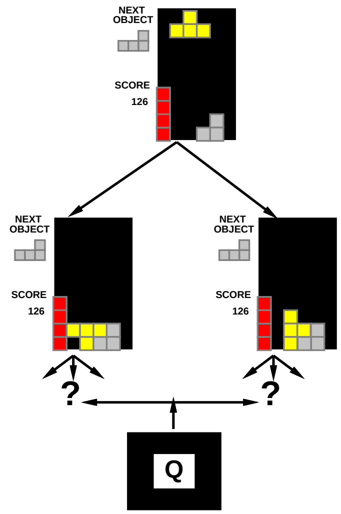
No time limit

$\Longrightarrow$ RL converges* to CR/BO

Time limit for whole game

$\Longrightarrow$ [convergence theorem here]

Metalevel design



Lookahead search controlled by metalevel Q-function

Q is a *fixed-runtime* function approximator

No time limit

$\Longrightarrow$ exhaustive search (degenerate Q)

gives BO/CR agent

Time limit

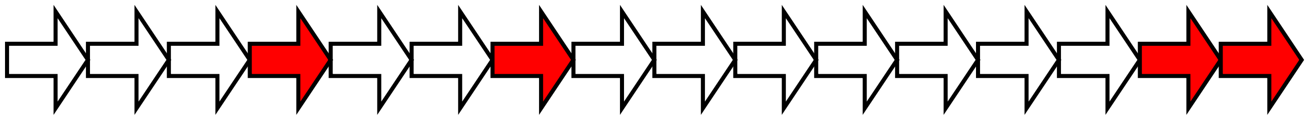
Optimal allocation of search resources is intractable

Need to learn a good approximate metalevel Q-function

Metalevel reinforcement learning

What are the rewards for computations?

Formally: construct MDP with joint internal/external states;
external actions are determined by internal computations



Case 1: external rewards only (checkmate)—slow convergence

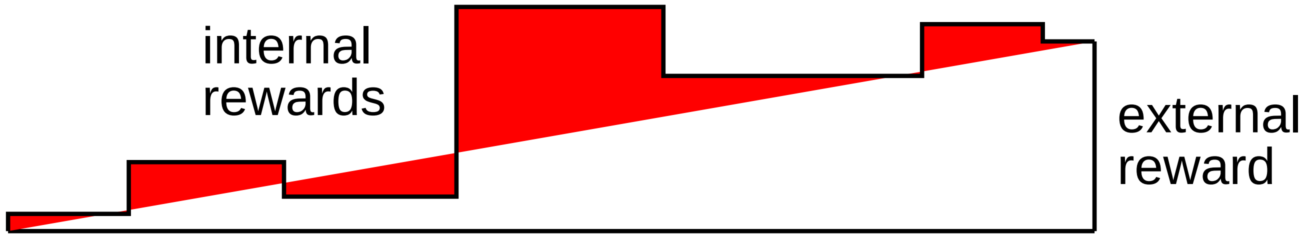
Case 2: reward for external action selection

Case 3: reward for each improved decision

Metalevel reinforcement learning contd.

"Fictitious" internal rewards must sum to real rewards

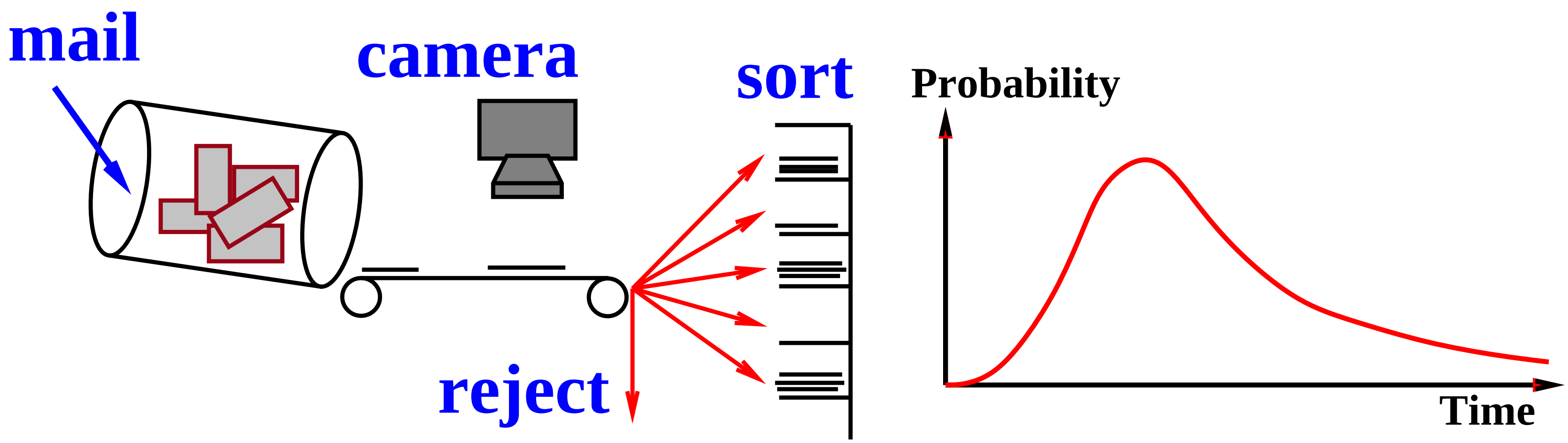
Arrange by rewarding change in *post hoc* value:



No net reward for selecting the original default choice!

Conjecture: given fixed object-level V, converges to BO

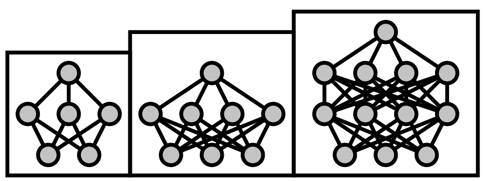
Composite systems



E: Letters arrive at random times

M: Runs one or more neural networks

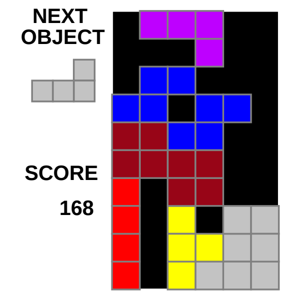
Can compute p_{opt}: a *sequence* of networks



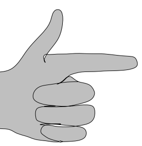
δ, ϵ-learned networks $\Rightarrow \delta', \epsilon'$-BO agent

Multiple execution architectures

Often need to combine e.g. reactive
and lookahead designs



Intuitively, should prove *dominance* of
combined architecture over either component



Dominance results need a robust notion of optimality

Asymptotic bounded optimality

Strict bounded optimality is too fragile

p is *asymptotically bounded-optimal* (ABO) iff

$\exists k\ V(Agent(p, kM), \mathbf{E}) \geq V(Agent(p_{opt}, M), \mathbf{E})$

I.e., speeding up M by k compensates

for p's inefficiency

Worst-case ABO and average-case ABO

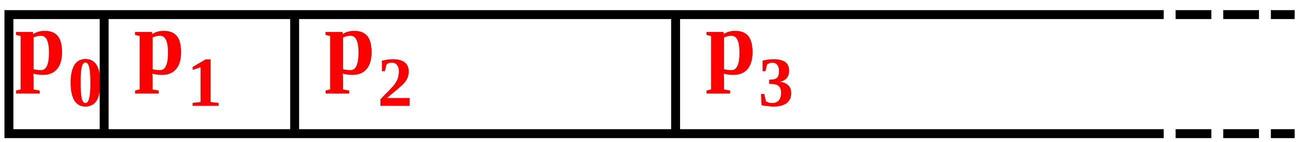
generalize classical complexity

Example: unknown deadlines

Suppose programs can be constructed easily for *fixed* deadlines

Let p_i be ABO for a fixed deadline at $t = 2^i \epsilon$

Construct the following universal program p_U

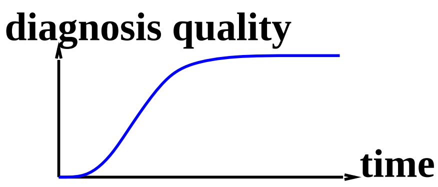


p_U is ABO for any deadline distribution

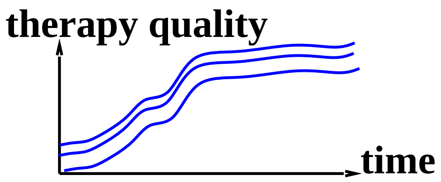
As good as knowing the deadline in advance.

Functional composition of systems

E.g., real-time diagnosis/therapy



therapy(*diagnose*(*x*))

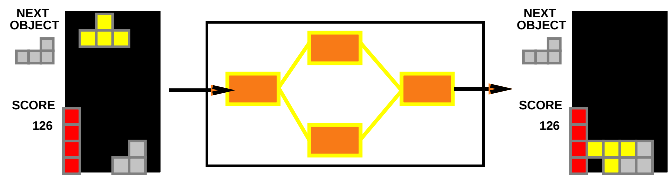


Theorem: given *input monotonicity* of profiles and *tree-structured* composition, optimal allocation of time with a fixed deadline can be computed in linear time

Composition with unknown deadlines

Use doubling construction to build *composite* anytime systems

ABO composite systems for unknown deadlines can be constructed in linear time



Composition language with loops, conditionals, logical expressions

$\Rightarrow$ "compiler" for complex systems

Need a more "organic" notion of composition

Radical bounded optimality

Several "forces" operate on agent configuration:

- ◇ Towards optimal decisions given current knowledge
- ◇ Towards instantaneous decisions
- ◇ Towards consistency with the environment

Complex architectures have several adaptation mechanisms:

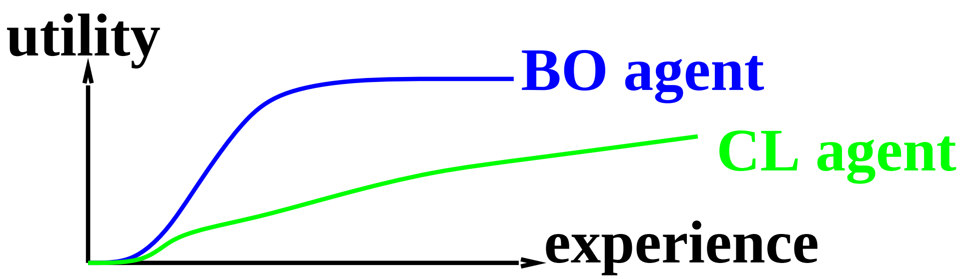
- ◇ Reinforcement learning (object- and meta-level)
- ◇ Model learning
- ◇ Compilation

Study agents as dynamical systems in dynamic quasiequilibrium with a changing environment

Learning is expensive

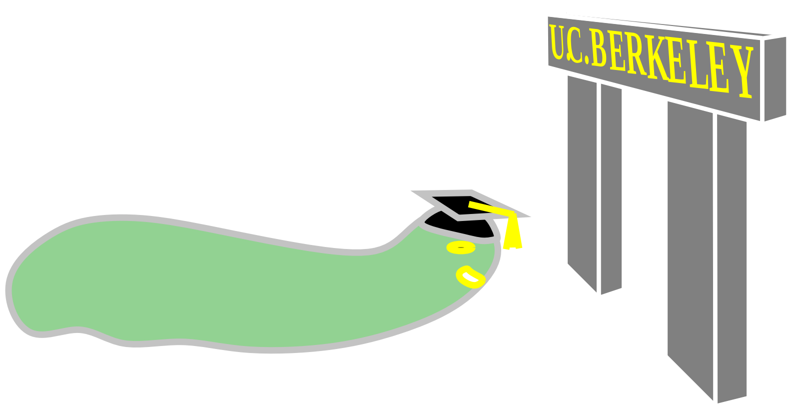
"Eventually converges to bounded optimality" is not enough

It's also too much: BO for a specific complex environment.
Less specification $\Rightarrow$ easier design problem

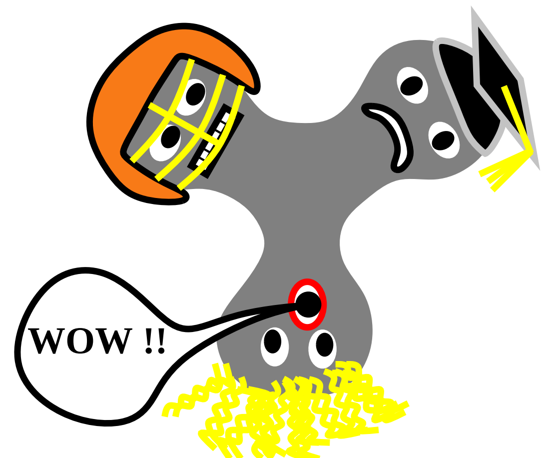


The complexity of BO agent design is not necessarily related to the complexity of the decisions the agent makes or to the complexity of the agent after learning.

Equilibrium configurations



Simple equilibrium



Complex equilibrium

Conclusions

- Computational limitations ~~Brains~~ cause minds
- Tools in, algorithms out (eventually)
- Bounded optimality:

    Fits intuitive idea of *Intelligence*

    A bridge between theory and practice
- Learning $\neq$ perfect modelling of environment
- Interesting architectures $\Rightarrow$ interesting learning behaviour