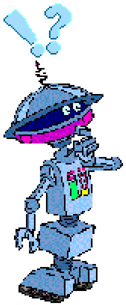


A Self-Made Agent Based on Action-Selection

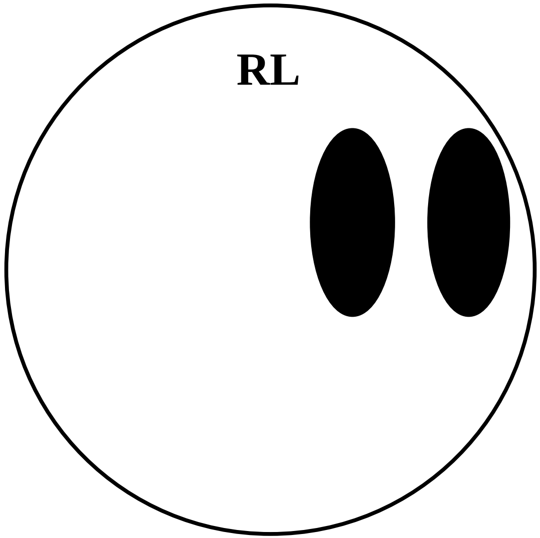
Olivier BUFFET
Alain DUTECH

Maia team - LORIA
INRIA

http://www.loria.fr/equipes/maia/

Problem

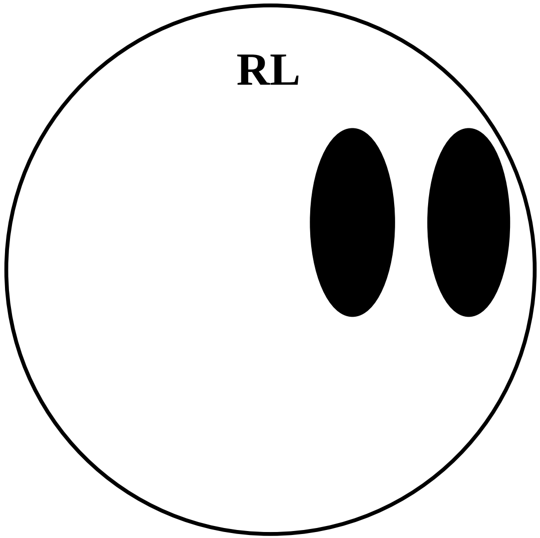
To automatically design agents
($\longrightarrow$ Reinforcement Learning)



Problem

To automatically design agents
(→ Reinforcement Learning)

facing several motivations
(what leads to low local optima).

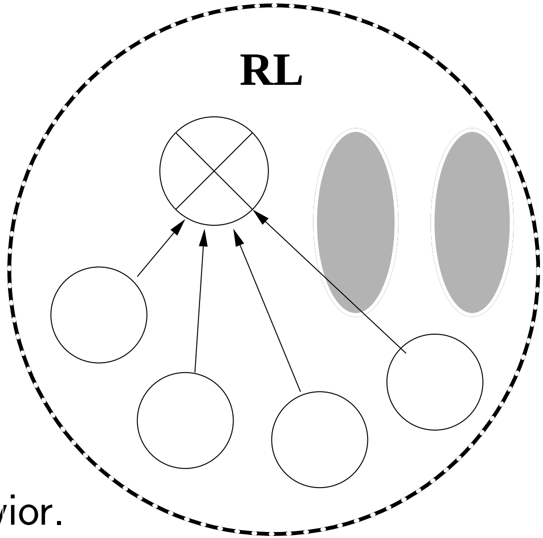


Problem

To automatically design agents
($\rightarrow$ Reinforcement Learning)

facing several motivations
(what leads to low local optima).

Idea: to decompose the agent's behavior.



Plan

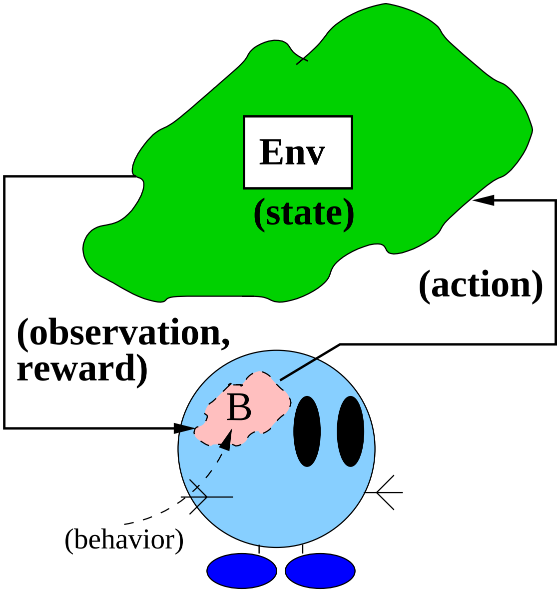
- Introduction

Simple Combination of Behaviors

Basic Behaviors Generation

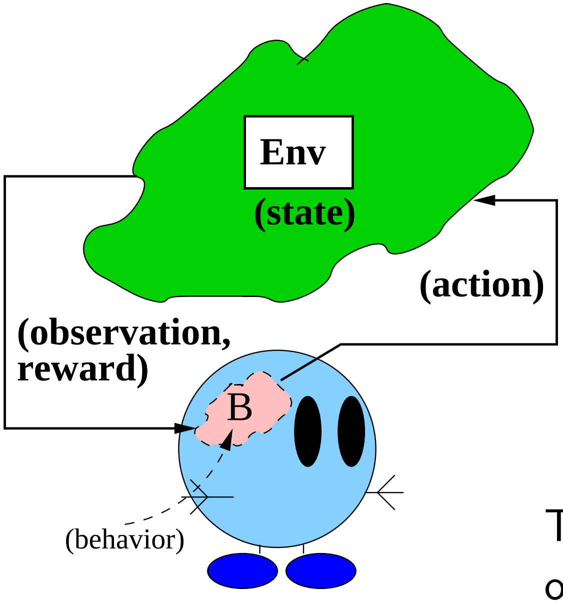
Conclusion & Open Questions

An agent (in this work)



- is in an uncertain environment,
- perceives it only partially, and
- acts on it while being motivated ($\longrightarrow$ reward).

An agent (in this work)



- is in an uncertain environment,
- perceives it only partially, and
- acts on it while being motivated ($\longrightarrow$ reward).

The problem of finding a behavior optimizing these rewards can be modeled as a POMDP.

Partially Observable Markovian Decision Processes (POMDP)

A POMDP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O} \rangle$

states	$\mathcal{S} = \{s\}$ (finite)
actions	$\mathcal{A} = \{a\}$ (finite)
transitions	$\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$
rewards	$\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
observations	$\Omega = \{o\}$ (finite)
obs. fct.	$\mathcal{O} : \mathcal{S} \rightarrow \Pi(\Omega)$

Partially Observable Markovian Decision Processes (POMDP)

A POMDP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O} \rangle$

states	$\mathcal{S} = \{s\}$ (finite)
actions	$\mathcal{A} = \{a\}$ (finite)
transitions	$\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$
rewards	$\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
observations	$\Omega = \{o\}$ (finite)
obs. fct.	$\mathcal{O} : \mathcal{S} \rightarrow \Pi(\Omega)$

Our agent

doesn't know any <u>model</u> of its environment $(\mathcal{S}, \mathcal{T}, \mathcal{R}, \mathcal{O})$ and has no short-term <u>memory</u>.

Partially Observable Markovian Decision Processes (POMDP)

A POMDP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O} \rangle$

states	$\mathcal{S} = \{s\}$ (finite)
actions	$\mathcal{A} = \{a\}$ (finite)
transitions	$\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$
rewards	$\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
observations	$\Omega = \{o\}$ (finite)
obs. fct.	$\mathcal{O} : \mathcal{S} \rightarrow \Pi(\Omega)$

Our agent

doesn't know any model of its environment $(\mathcal{S}, \mathcal{T}, \mathcal{R}, \mathcal{O})$ and has no short-term memory.

Goal

To equip the agent with a policy

$$\pi : \Omega \rightarrow \Pi(\mathcal{A})$$

maximising the expected long-term reward.

Existing RL algorithms

Can be cited:

- REINFORCE [Wil92],
- Jaakkola's Q-learning [JJS94],
- Baxter and Bartlett's gradient descent [BB01] ...

Existing RL algorithms

Can be cited:

- REINFORCE [Wil92],
- Jaakkola's Q-learning [JJS94],
- Baxter and Bartlett's gradient descent [BB01] ...

Some difficulties [**KLM96**]:

- large observation spaces,
- missing reward signal, and
- solutions which are only local optima.

Outline

Introduction

■ Simple Combination of Behaviors

Basic Behaviors Generation

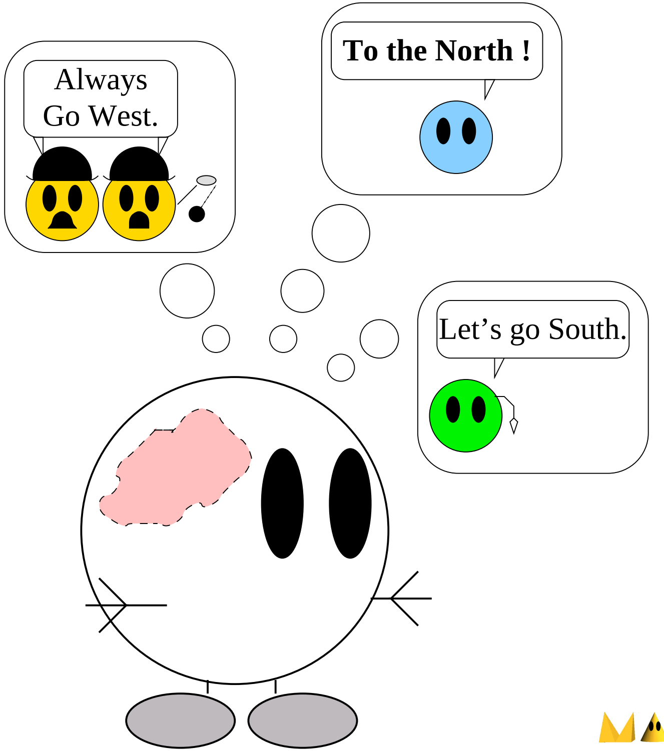
Conclusion & Open Questions

Principle of Action Selection

To be able to manage several motivations simultaneously. [Tyr93]

Principle of Action Selection

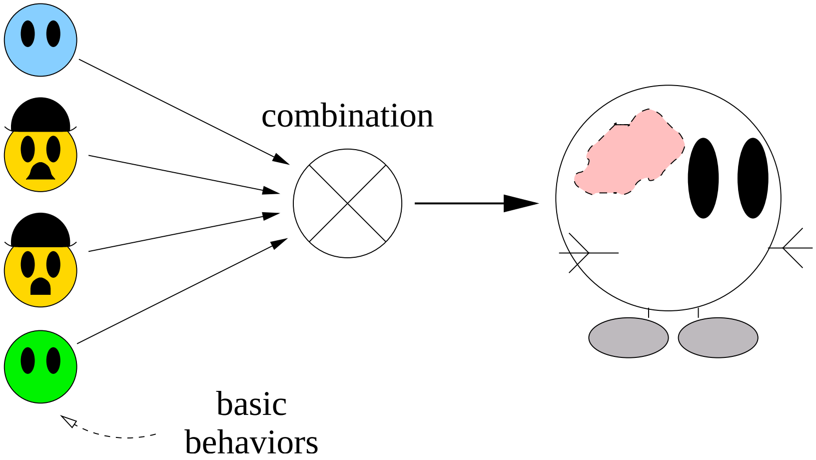
To be able to manage several motivations simultaneously. [Tyr93]



A classical architecture:

decision-making system using the votes of expert modules (each having its own motivation).

Our goal (in this framework)

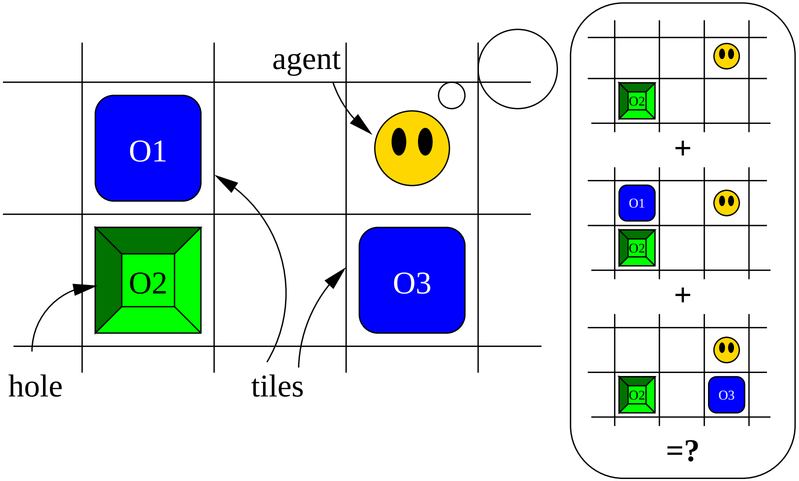


To automate the agent's conception, *i.e.*:

- the <u>creation</u> and <u>choice</u> of basic behaviors and
- the tuning of the <u>combination</u>.

Scene decomposition

(finding the motivations)



$(behavior, config)$ pairs:

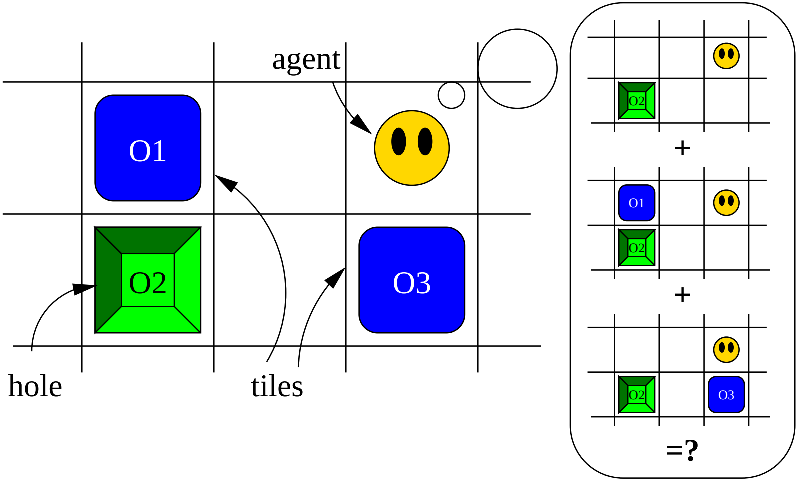
$(b_e, \{O_2\})$

$(b_p, \{O_2, O_1\})$

$(b_p, \{O_2, O_3\})$

Scene decomposition

(finding the motivations)



$(behavior, config)$ pairs:

$(b_e, \{O_2\})$

$(b_p, \{O_2, O_1\})$

$(b_p, \{O_2, O_3\})$

Typed basic behaviors:

avoid hole	$(b_e, \{\langle \text{hole} \rangle\})$
push tile	$(b_p, \{\langle \text{hole} \rangle, \langle \text{tile} \rangle\})$

Basic Behaviors (bb)

A generic basic behavior b is defined by

a type of configuration	$\{\langle T_{obj\ 1}\rangle, \ldots, \langle T_{obj\ k}\rangle\}$,
a stochastic policy	$P_b(a\vert o(c))$,
and the corresponding Q-table	$Q_b(a, o(c))$.

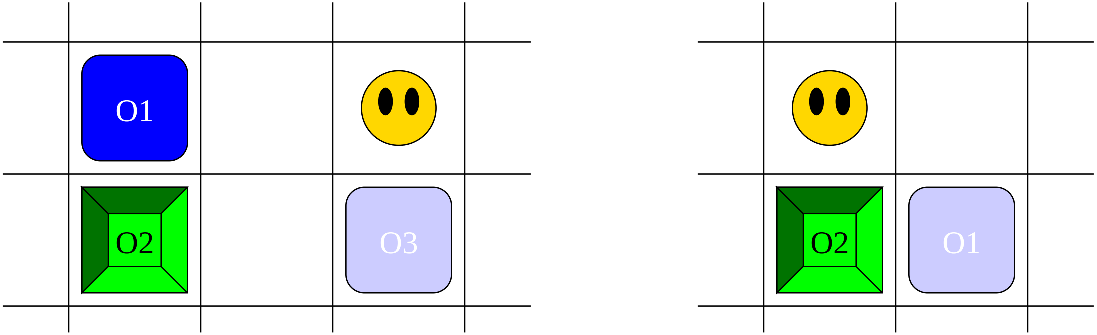
Note: P_b and Q_b are learned beforehand.

Basic Behaviors (*bb*)

A generic basic behavior b is defined by

a type of configuration	$\{\langle T_{obj\ 1}\rangle, \ldots, \langle T_{obj\ k}\rangle\}$,
a stochastic policy	$P_b(a\|o(c))$,
and the corresponding Q-table	$Q_b(a, o(c))$.

Note: P_b and Q_b are learned beforehand.



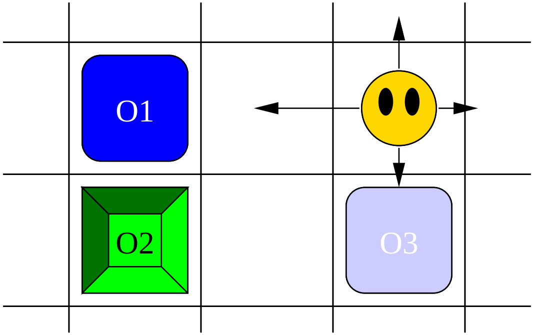
push tile in hole — *avoid* hole

Basic Behaviors (bb)

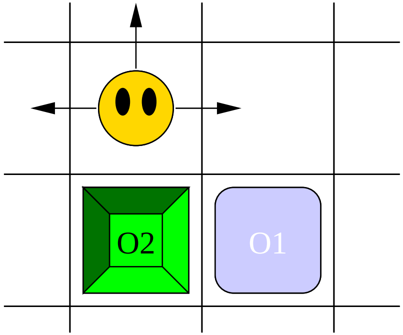
A generic basic behavior b is defined by

a type of configuration	$\{\langle T_{obj\ 1}\rangle, \ldots, \langle T_{obj\ k}\rangle\}$,
a stochastic policy	$P_b(a\|o(c))$,
and the corresponding Q-table	$Q_b(a, o(c))$.

Note: P_b and Q_b are learned beforehand.



push tile in hole



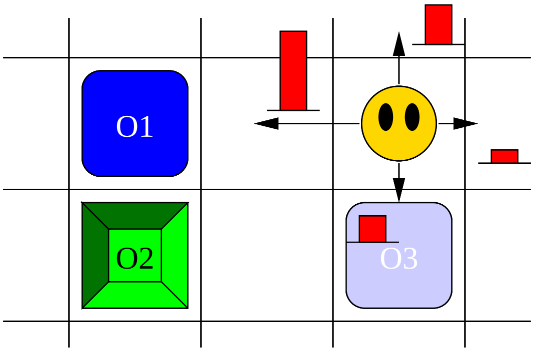
avoid hole

Basic Behaviors (*bb*)

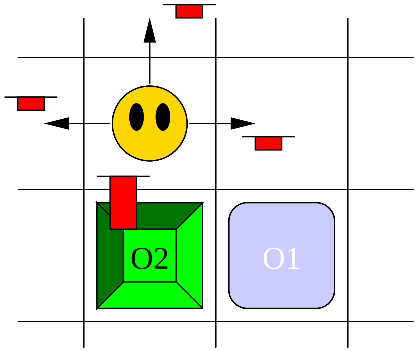
A generic basic behavior b is defined by

a type of configuration	$\{\langle T_{obj\ 1}\rangle, \ldots, \langle T_{obj\ k}\rangle\}$,
a stochastic policy	$P_b(a\|o(c))$,
and the corresponding Q-table	$Q_b(a, o(c))$.

Note: P_b and Q_b are learned beforehand.



push tile in hole



avoid hole

Basic Behaviors Combination

Each basic behavior $\longrightarrow$ One policy: $P_b(a|o(c))$

Basic Behaviors Combination

	Each basic behavior	$\longrightarrow$	One policy: $P_b(a\|o(c))$
Goal	Several basic behaviors	$\longrightarrow$	One complex policy

Basic Behaviors Combination

	Each basic behavior	$\longrightarrow$	One policy: $P_b(a\|o(c))$
Goal	Several basic behaviors	$\longrightarrow$	One complex policy

An example among the proposed formulas

$$\mathcal{P}(a|o) = \frac{1}{K} \sum_{(b,c)\in\mathcal{BC}(o)} e^{\zeta_b}.|Q_b(a,o(c))|.P_b(a|o(c))$$

Basic Behaviors Combination

	Each basic behavior	$\longrightarrow$	One policy: $P_b(a\|o(c))$
Goal	Several basic behaviors	$\longrightarrow$	One complex policy

An example among the proposed formulas

$$\mathcal{P}(a|o) = \frac{1}{K} \sum_{(b,c)\in\mathcal{BC}(o)} e^{\zeta_b}.|Q_b(a,o(c))|.P_b(a|o(c))$$

normalization

Basic Behaviors Combination

	Each basic behavior	$\longrightarrow$	One policy: $P_b(a\|o(c))$
Goal	Several basic behaviors	$\longrightarrow$	One complex policy

An example among the proposed formulas

$$\mathcal{P}(a|o) = \frac{1}{K} \sum_{(b,c)\in\mathcal{BC}(o)} e^{\zeta_b}.|Q_b(a,o(c))|.P_b(a|o(c))$$

normalization

each (bb,cfg) pair is taken into account

Basic Behaviors Combination

	Each basic behavior	$\longrightarrow$	One policy: $P_b(a\|o(c))$
Goal	Several basic behaviors	$\longrightarrow$	One complex policy

An example among the proposed formulas

$$\mathcal{P}(a|o) = \frac{1}{K} \sum_{(b,c)\in\mathcal{BC}(o)} e^{\zeta_b}.|Q_b(a,o(c))|.\underbrace{P_b(a|o(c))}_{\text{probability to choose a}}$$

normalization

each (bb,cfg) pair is taken into account

probability to choose a

Basic Behaviors Combination

	Each basic behavior	$\longrightarrow$	One policy: $P_b(a\|o(c))$
Goal	Several basic behaviors	$\longrightarrow$	One complex policy

An example among the proposed formulas

$$\mathcal{P}(a|o) = \frac{1}{K} \sum_{(b,c)\in\mathcal{BC}(o)} e^{\zeta_b}.\underbrace{|Q_b(a,o(c))|}_{\text{quality of this decision}}.\underbrace{P_b(a|o(c))}_{\text{probability to choose a}}$$

normalization

each (bb,cfg) pair is taken into account

Basic Behaviors Combination

	Each basic behavior	$\longrightarrow$	One policy: $P_b(a\|o(c))$
Goal	Several basic behaviors	$\longrightarrow$	One complex policy

An example among the proposed formulas

$$\mathcal{P}(a|o) = \frac{1}{K} \sum_{(b,c)\in\mathcal{BC}(o)} e^{\zeta_b} . \underbrace{|Q_b(a, o(c))|}_{\text{quality of this decision}} . \underbrace{P_b(a|o(c))}_{\text{probability to choose a}}$$

normalization

each (bb,cfg) pair is taken into account

weight each bb linked to

Utility of ζ_b

The weighting parameters ζ_b have to be learned in order to equilibrate the behaviors' combination. They make it possible to:

Utility of ζ_b

The weighting parameters ζ_b have to be learned in order to equilibrate the behaviors' combination. They make it possible to:

Find a compromise

Reducing the importance of a behavior (ex.: *avoid*) can help another one working (ex.: *push*).

Utility of ζ_b

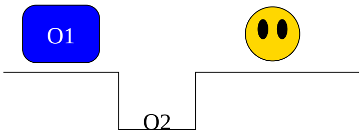
The weighting parameters ζ_b have to be learned in order to equilibrate the behaviors' combination. They make it possible to:

Find a compromise

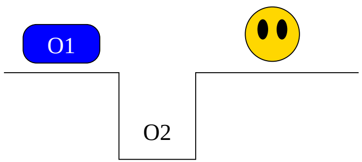
Reducing the importance of a behavior (ex.: *avoid*) can help another one working (ex.: *push*).

Change the rules of the game

Reuse basic behaviors in new conditions.



push	$+3$
avoid	-1



push	$+1$
avoid	-2

Intermediate weighing-up

Interesting results...

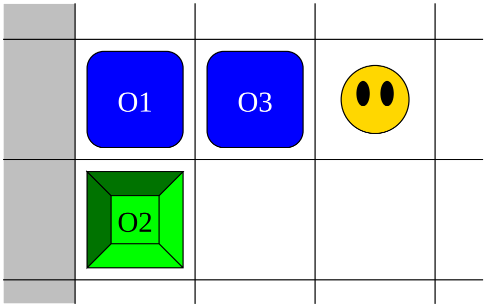
- local optima overcomed,
- good scalability,
- possible reuse of bbs.

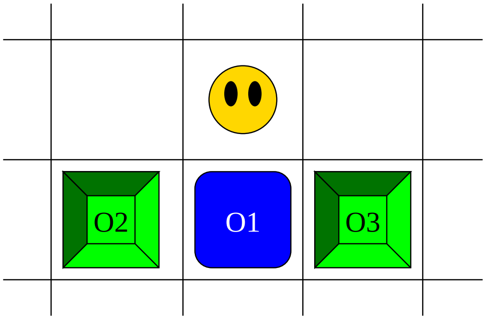
Intermediate weighing-up

Interesting results...

- local optima overcomed,
- good scalability,
- possible reuse of *bb*s.

...but also blockings



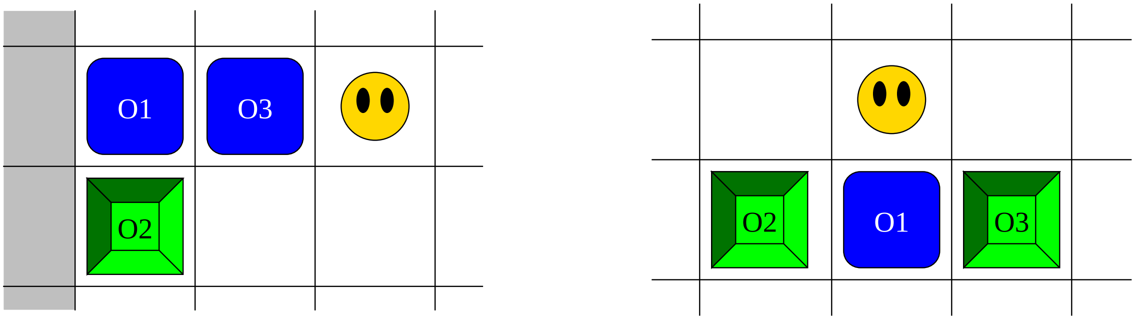


Intermediate weighing-up

Interesting results...

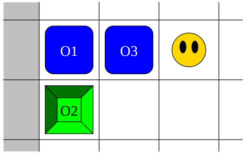
- local optima overcomed,
- good scalability,
- possible reuse of bbs.

...but also blockings



These bad decisions lead to an important loss in efficiency...

What should be done ?



Learn a new behavior

[concerns a type of configuration]

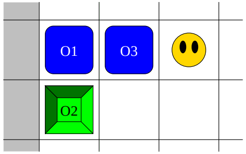
- May give a lot of work.

+ Will be reusable in new combinations.

+ Can be based on the faulty combined behavior (most decisions are pretty good ones).

What should be done ?

Learn a new behavior



[concerns a type of configuration]

- May give a lot of work.

+ Will be reusable in new combinations.

+ Can be based on the faulty combined behavior (most decisions are pretty good ones).

On this principle, the agent could build any new basic behavior which appears to be of interest.

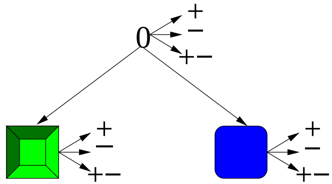
A progressive exploration of potential *bb*s could then lead to generate them all.

Outline

Tree growth

(exploring a tree of potential bbs)

Start:

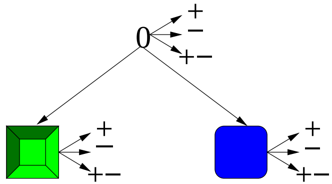


Simple situations to experience
in "virtual" environments.
No basic behaviors.

Tree growth

(exploring a tree of potential bbs)

Start:



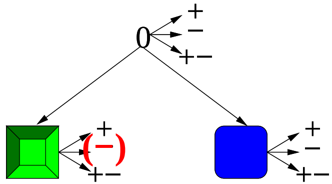
Simple situations to experience
in “virtual” environments.
No basic behaviors.

Without objects (or at most one tile): nothing to learn.
The combination without behaviors is sufficient
(it gives a random policy).

Tree growth

(exploring a tree of potential bbs)

Start:



Simple situations to experience
in "virtual" environments.
No basic behaviors.

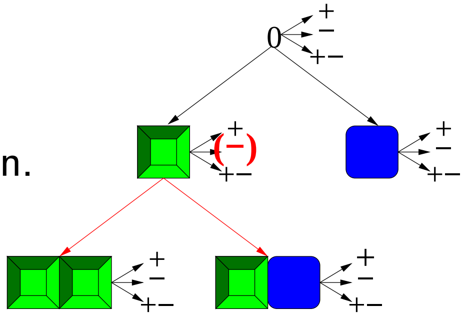
Without objects (or at most one tile): nothing to learn.
The combination without behaviors is sufficient
(it gives a random policy).

The agent only learns how to avoid a hole.

Tree growth

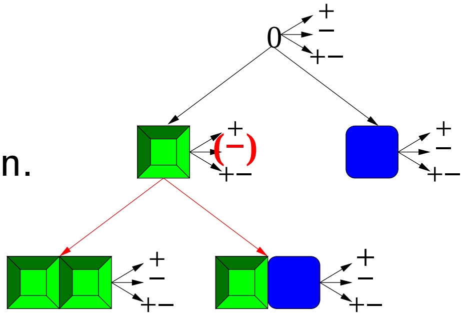
1^{st} development:

A sub-tree is developped for exploration.

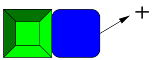


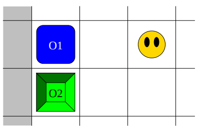
Tree growth

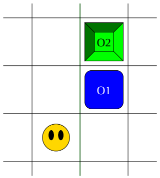
1^{st} development:

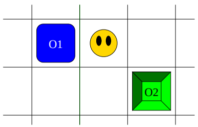


A sub-tree is developped for exploration.

The  symbol represents
1 type of configuration and
1 combination of rewards.

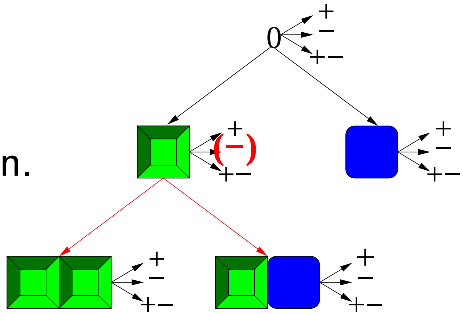




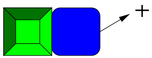


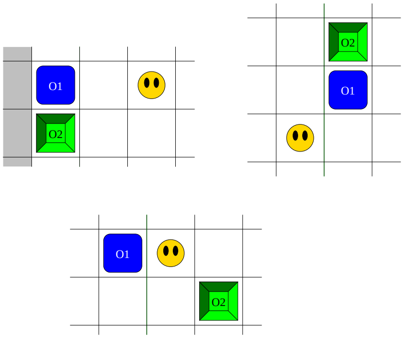
Tree growth

1^{st} development:



A sub-tree is developped for exploration.

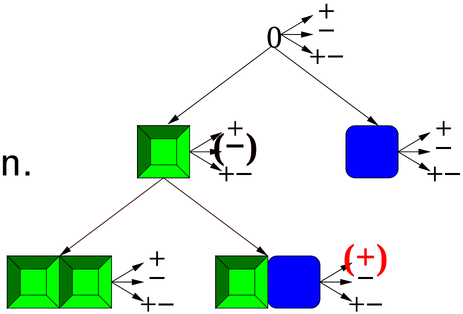
The  symbol represents
1 type of configuration and
1 combination of rewards.



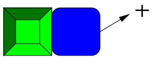
The experimentation compares here:
- a combination whose only bb avoids a hole, and
- a policy learned based on this combination.

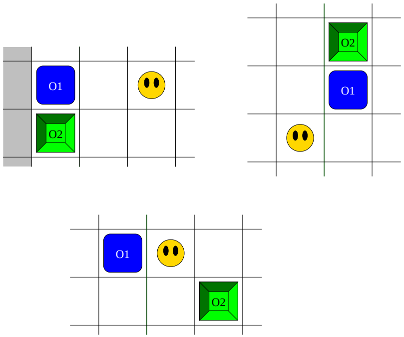
Tree growth

1^{st} development:



A sub-tree is developped for exploration.

The  symbol represents
1 type of configuration and
1 combination of rewards.

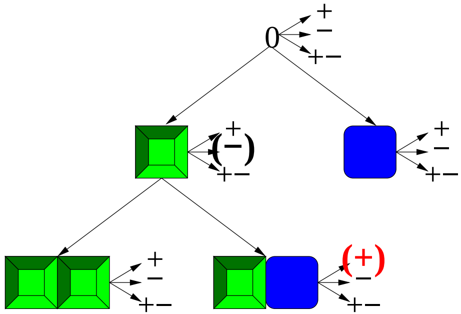


The experimentation compares here:
- a combination whose only
bb avoids a hole, and
- a policy learned based
on this combination.
$\longrightarrow$ application of a criterion

Tree growth

Retained basic behaviors:

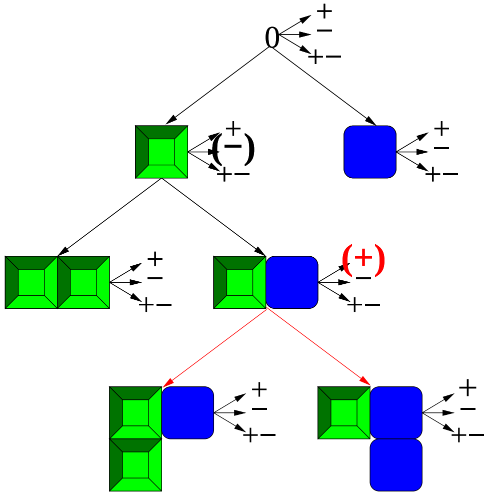
b_e : $(\{\langle \mathsf{hole} \rangle\},\ -)$
b_p : $(\{\langle \mathsf{hole} \rangle, \langle \mathsf{tile} \rangle\},\ +)$



Tree growth

Retained basic behaviors:

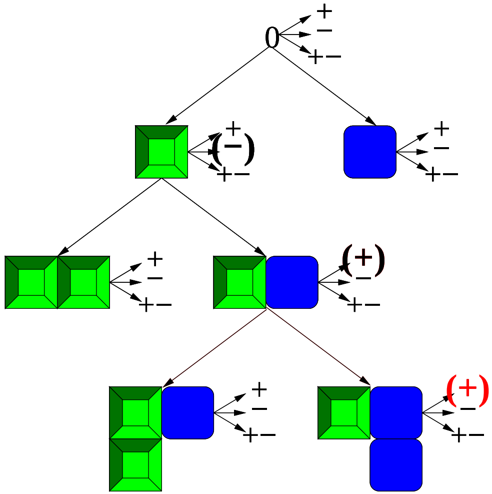
b_e : $(\{\langle\mathsf{hole}\rangle\},\ -)$
b_p : $(\{\langle\mathsf{hole}\rangle, \langle\mathsf{tile}\rangle\},\ +)$



Tree growth

Retained basic behaviors:

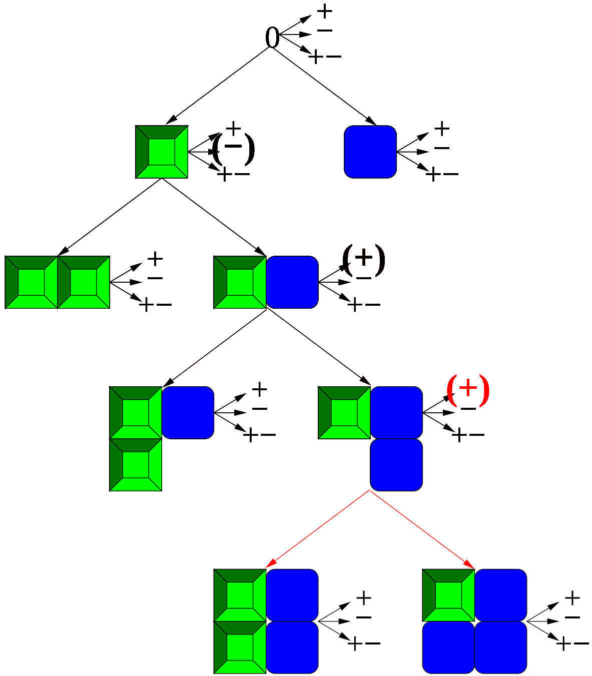
b_e : $(\{\langle\mathsf{hole}\rangle\},\ -)$
b_p : $(\{\langle\mathsf{hole}\rangle, \langle\mathsf{tile}\rangle\},\ +)$
$b_{p'}$: $(\{\langle\mathsf{hole}\rangle, \langle\mathsf{tile}\rangle, \langle\mathsf{tile}\rangle\},\ +)$



Tree growth

Retained basic behaviors:

b_e : $(\{\langle \text{hole} \rangle\},\ -)$
b_p : $(\{\langle \text{hole} \rangle, \langle \text{tile} \rangle\},\ +)$
$b_{p'}$: $(\{\langle \text{hole} \rangle, \langle \text{tile} \rangle, \langle \text{tile} \rangle\},\ +)$



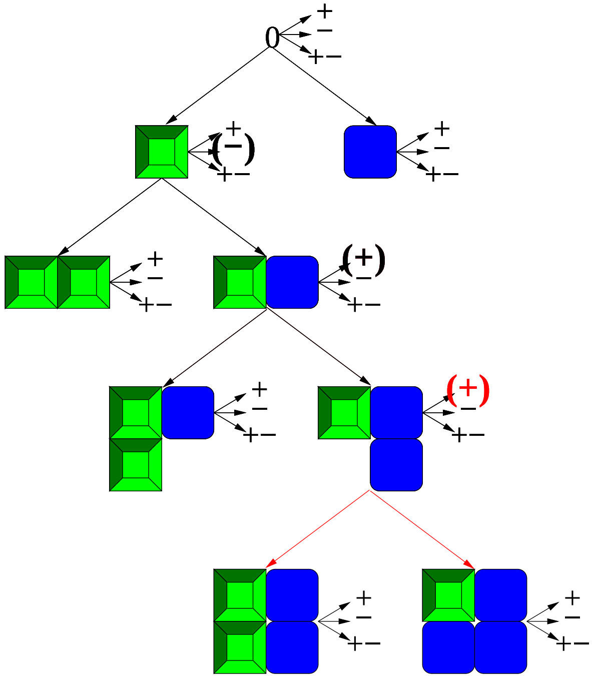
Tree growth

Retained basic behaviors:

b_e : $(\{\langle\text{hole}\rangle\},\ -)$
b_p : $(\{\langle\text{hole}\rangle, \langle\text{tile}\rangle\},\ +)$
$b_{p'}$: $(\{\langle\text{hole}\rangle, \langle\text{tile}\rangle, \langle\text{tile}\rangle\},\ +)$

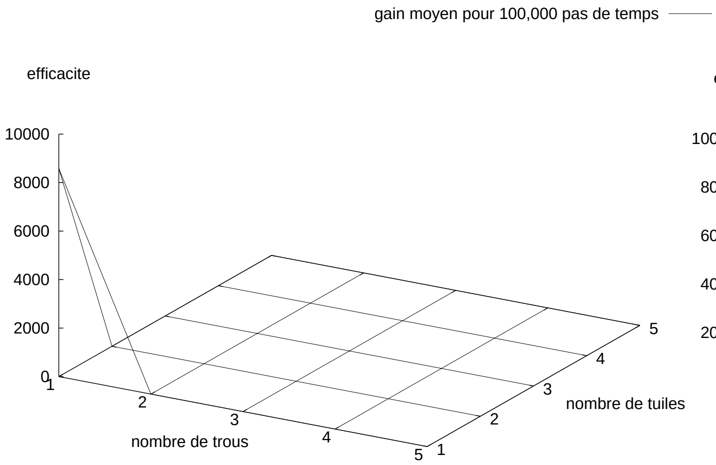
Result:

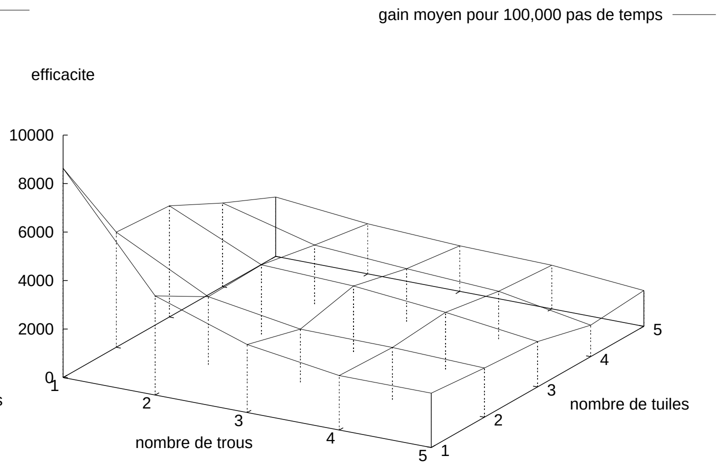
both "intuitive" bbs and
a bb solving one of the blockings.



Results

Comparaison of the average reward obtained between:





a classical RL algorithm, and our "autonomous" architecture.

Outline

Introduction

Simple Combination of Behaviors

Basic Behaviors Generation

■ Conclusion & Open Questions

Conclusion...

Conclusion...

Simple combination of Behaviors

- local optima overcomed,
- good scalability, and
- possible reuse of bbs.

Conclusion...

Simple combination of Behaviors

- local optima overcomed,
- good scalability, and
- possible reuse of bbs.

Basic Behavior Generation

- finds both intuitive behaviors $+$ 1 "blocking-solver",
- but depends on key-parameters (retaining criterion, depths...).

Conclusion...

Simple combination of Behaviors

- local optima overcomed,
- good scalability, and
- possible reuse of bbs.

Basic Behavior Generation

- finds both intuitive behaviors + 1 “blocking-solver”,
- but depends on key-parameters (retaining criterion, depths...).

This Action-Selection system is autonomous and still efficient when classical RL does not work.

...& Open Questions

Applications:

- path planning + avoidance,
- resource management.

Can We Find a Better Method of Recombination ?

Characterization of relationships between basic behaviors (when b_1 and b_2 are concurrent, if b_3 inhibits b_4...)
$\longrightarrow$ more appropriate recombinations

Extraction of Abstract Object Types

Both a tile or a hole could be considered in some situations as being "obstacles".

Bibliography

[BB01] J. Baxter and P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[JJS94] T. Jaakkola, M. Jordan, and S. Singh. Reinforcement learning algorithm for partially observable markov decision problems. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 345–352. The MIT Press, 1994.

[KLM96] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[Tyr93] T. Tyrrell. *Computational Mechanisms for Action Selection*. PhD thesis, University of Edinburgh, 1993.

[Wil92] R.J. Williams. Simple statistical gradient-following algorithms for connectionnist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

Time for questions...

[insert questions here]