



UNIVERSITÀ
DI TORINO

INTELLIGENZA ARTIFICIALE & LABORATORIO

ESCAPE GAME

Progetto SOAR - Parte n.3 - Prof. Lieto

Data: 15 Settembre 2023

Delmastro Andrea (912954) - Ferrero Fabio (926392) - Frumento Giulia (834773)

Università degli Studi di Torino

INDICE

1. Escape Game
2. Modellazione della conoscenza
3. Assunzioni e semplificazioni del problema
4. Impasse
5. Rewards
6. Performance

ESCAPE GAME

INTRODUZIONE ESCAPE GAME

In questo progetto viene utilizzata l'architettura cognitiva SOAR al fine di modellare il comportamento di un agente (e.g. *un robot*) confinato all'interno di un ambiente (e.g. una *stanza*), con l'obiettivo di fuggire attraverso una finestra blindata.

L'agente ha a disposizione diversi oggetti, di cui tuttavia all'inizio non comprende il funzionamento. Ciò nonostante, attraverso l'applicazione di un meccanismo di Apprendimento per Rinforzo (Reinforcement Learning - RL), l'agente acquisirà la capacità di combinare adeguatamente gli oggetti per creare strumenti, frantumare la finestra, e riuscire a fuggire.

MODELLAZIONE DELLA CONOSCENZA

MODELLAZIONE DELLA CONOSCENZA: **AMBIENTE, AGENTE E MOVIMENTO**

Sono state effettuate le seguenti operazioni di modellazione della conoscenza

- L'agente si trova in un ambiente costituito da una stanza con quattro pareti disposte lungo i punti cardinali (nord, sud, est, ovest). All'inizio, la posizione dell'agente è al centro esatto della stanza. Una delle pareti ospita una finestra ad altezza $3,5m$. L'agente è alto $1,5m$.
Il punto **est** è stato selezionato come la posizione della finestra.
- Le possibilità di movimento dell'agente comprendono spostamenti nelle direzioni **nord, sud, est e ovest**. Ogni movimento richiede che l'agente attraversi sempre la posizione centrale.
- All'agente manca la conoscenza sulla posizione esatta della finestra, pertanto dovrà individuarla attraverso la ricerca.
- La finestra ha uno **stato di condizione** che va da 0% a 100%. La finestra di considera frantumata quando il suo stato è $\leq 0\%$.

MODELLAZIONE DELLA CONOSCENZA: **OGGETTI E COSTRUZIONE DI STRUMENTI**

- L'agente ha a disposizione la seguente collezione di oggetti:
Elastico (Rubber-band), **Rametto di legno** (Sprig) e **Ciottoli** (Pebble).
- L'agente può **costruire strumenti**, combinando gli oggetti che possiede. Nello specifico:
 1. **Elastico + Rametto = Fionda** (Slingshot)
 2. **Elastico + Ciottoli = Strumento inutile 1** (Silly tool 1)
 3. **Ciottoli + Rametto = Strumento inutile 2** (Silly tool 2)

Gli oggetti impiegati per fabbricare lo strumento vengono sottratti dalla collezione dell'agente, e il nuovo strumento viene inserito.
- L'agente ha la capacità di **distuggere** gli strumenti prodotti, riottenendo così gli oggetti impiegati nella loro creazione.

MODELLAZIONE DELLA CONOSCENZA: **LANCIARE** E **SPARARE**

- L'agente può colpire la finestra mediante le seguenti due azioni:
 1. **LANCIARE** (Throw) un **oggetto** o uno **strumento** creato.
 2. **SPARARE** (Shot) il **ciottolo** con la **fionda**.
- Ogni volta che viene effettuato un **lancio**, il valore dello stato della finestra subirà una riduzione corrispondente alla potenza dell'oggetto/strumento lanciato. I valori della potenza degli oggetti/strumenti sono:
 - Elastico: $\frac{1}{50}\%$
 - Rametto di legno: $\frac{1}{20}\%$
 - Ciottolo: $\frac{1}{2}\%$
 - Fionda = Strumento inutile 1 = Strumento inutile 2 = 1%

MODELLAZIONE DELLA CONOSCENZA: **LANCIARE E SPARARE**

- Gli **spari** di un ciottolo con la fionda possono colpire la finestra in due punti differenti, ovvero
 - al **CENTRO**: in questa situazione, la potenza del colpo sparato varia (in modo casuale) dal 1% al 5%.
 - agli **ANGOLI**: in questa situazione, la potenza del colpo sparato varia (in modo casuale) dal 50% a 150%.
- All'agente manca la conoscenza dell'azione ottimale da intraprendere. In altre parole, non è consapevole se sia più vantaggioso creare un determinato strumento rispetto un altro, così come non sa se sia preferibile lanciare un oggetto/strumento piuttosto che un altro.
Inizialmente, l'agente non possiede la conoscenza se sia più conveniente mirare al centro o ai bordi della finestra quando utilizza la fionda.

MODELLAZIONE DELLA CONOSCENZA: **ESEMPI DI LANCIO E SPARO**

Esempi (sia w_c lo stato della finestra e ps la potenza del colpo dello sparo con la fionda)

- $w_c = 100\%$. **Lancio** del **Rametto di legno** $\Rightarrow w_c = 100 - \frac{1}{20} = 99.95\%$
- $w_c = 80\%$. **Lancio** della **fionda** $\Rightarrow w_c = 80 - 1\frac{1}{20} = 79\%$
- $w_c = 70\%$. **Sparo** con la fionda con $ps = 80\%$ $\Rightarrow w_c = 70 - 80 = -10\%$.
Finestra frantumata.

MODELLAZIONE DELLA CONOSCENZA: **COSTRUZIONE SCALA E USCITA**

- All'interno della stanza sono presenti due tronchi con le seguenti dimensioni
 - Altezza: 1 metro;
 - Diametro: 30 centimetri;
- L'agente può **costruire una scala** utilizzando i due tronchi. Più precisamente, è possibile realizzare la scala utilizzando due approcci distinti
 1. Disposizione **orizzontale**: i due tronchi vengono sovrapposti orizzontalmente per creare una scala di $30cm + 30cm = 60cm$;
 2. Disposizione **verticale**: i due tronchi vengono sovrapposti orizzontalmente per creare una scala di $1m + 1m = 2m$;
- Inizialmente, l'agente **non** possiede informazioni su quale scala sia più idonea da costruire per riuscire a sfuggire dalla stanza
- Nel caso in cui l'agente eriga una scala la cui lunghezza, sommata all'altezza dell'agente stesso, sia \geq dell'altezza della finestra, si deduce che il robot è in grado di fuggire dall'ambiente.

ASSUNZIONI E SEMPLIFICAZIONI DEL PROBLEMA

ASSUNZIONI E SEMPLIFICAZIONI DEL PROBLEMA

Sono state effettuate le seguenti assunzioni al fine di evitare eccessiva complessità del problema.

1. Si ipotizza che gli oggetti a disposizione dell'agente non siano posizionati sul pavimento della stanza, ma che egli li abbia con sé;
2. Dopo aver individuato la posizione della finestra (inizialmente sconosciuta all'agente), si suppone che i due tronchi si trovino nella medesima posizione della finestra stessa.
3. Si presume che l'agente sia a conoscenza del fatto che, prima di procedere con la combinazione degli oggetti o di eseguire uno sparo o un lancio, è necessario che abbia individuato la finestra.
4. È ipotizzato che l'agente sia al corrente del fatto che, avendo accesso alla fionda, è più opportuno impiegarla per sparare il ciottolo anziché semplicemente lanciare un oggetto.

IMPASSE

IMPASSE

- **Impasse per il MOVIMENTO (nord-sud-est-ovest)**

Inizialmente, l'operatore viene selezionato casualmente (*numeric indifferent* NI), successivamente la scelta avviene attraverso l'apprendimento per rinforzo (RL). Qualora l'agente si trovi in una posizione diversa rispetto a quella della finestra, il suo unico movimento possibile sarà il ritorno al centro.

- **Impasse per la CREAZIONE e DISTRUZIONE di strumenti**

All'inizio, l'operatore è scelto in modo casuale (*numeric indifferent*); in seguito, la selezione avviene tramite RL.

- **Impasse per LANCIO di un certo oggetto/strumento alla finestra**

Anche in questo caso, all'inizio la scelta dell'operatore è casuale (NI) e successivamente la selezione si verifica mediante RL.

IMPASSE

- **Impasse per lo SPARO con la fionda al CENTRO o BORDI della finestra**

Inizialmente NI, successivamente la scelta avviene attraverso il Reinforcement Learning.

È importante notare che, una volta individuata la finestra, l'agente ha la flessibilità di decidere se combinare oggetti, distruggere strumenti, sparare o lanciare, senza un ordine predeterminato. A titolo di esempio, l'agente potrebbe optare per la costruzione della fionda, per poi distruggerla immediatamente. In alternativa, potrebbe decidere di creare uno strumento e, invece di impiegarlo, lanciare un oggetto verso la finestra.

- **Impasse per la CREAZIONE e DISTRUZIONE della scala ORIZZONTALE/VERTICALE**

Inizialmente *numeric indifferent*, successivamente RL.

REWARDS

REWARDS

Action	Reward
move-est	+1
move-nord	-1
move-sud	-1
move-ovest	-1
combine-slingshot	+1
combine-silly-tool- (i)	-1
destroy-slingshot	-2
destroy-silly-tool- (i)	-1

Action	Reward
throw	$+\frac{1}{100}$
shot-center	$+\frac{1}{10}$
shot-edge	+1
create-ladder-horiz	-1
create-ladder-vertic	+1
destroy-ladder	-1

PERFORMANCE

PERFORMANCE

Come specificato nel manuale SOAR (versione 9.6.0), il comando `indifferent-selection` permette di determinare come vengono selezionati gli operatori in base alla loro preferenza `numeric-indifferent`.

Come riportato nel manuale nel Capitolo 5.3.3.1, le due strategie di esplorazione più comuni sono `epsilon-greedy` e `boltzmann`. La strategia di default è `softmax`.

Per questo progetto, è stato scelto l'approccio `epsilon-greedy` al fine di esaminare le esecuzioni al variare del valore di *epsilon* configurato.

Valutazione delle performance

Per ciascun valore di ϵ preso in considerazione, sono stati condotti 10 test. In ognuno di questi test, l'agente SOAR è stato eseguito fino al raggiungimento del numero minimo di azioni richieste (ovvero [5, 6]) per uscire dall'ambiente, con un limite massimo di 10 tentativi.

PERFORMANCE

- N : il numero medio di esecuzioni dell'agente SOAR;
- Opt : il conteggio delle occasioni in cui, su 10 test effettuati, l'agente è riuscito a compiere il numero minimo di azioni necessarie per fuggire dall'ambiente.
- min : la media del numero minimo di azioni compiute dall'agente nei 10 test.
- max : la media del numero massimo di azioni compiute dall'agente nei 10 test.

Valore di epsilon	N	Opt	min	max
$\epsilon = 0,1$	2,3	10/10	5.8	6.4
$\epsilon = 0,3$	3,1	10/10	5.5	7.9
$\epsilon = 0,5$	4,5	10/10	5.6	13.6
$\epsilon = 0,7$	8,8	4/10	6.8	30.8
$\epsilon = 0,9$	9,2	1/10	9.3	53.3