

# Relazione progetto di Intelligenza Artificiale e Laboratorio Modulo: Planning e Sistemi a Regole

Andreatta Chiara, Gerbaudo Luca, Racca Andrea Rocco  
a.a. 2019 - 2020

## Introduzione

In questa relazione è illustrata l'implementazione di un sistema esperto, in grado di dedurre la posizione del maggior numero possibile di navi del gioco Battaglia Navale. Nello sviluppo di tale sistema abbiamo scelto di partire da un primo Agente molto semplice, che è stato via via migliorato.

La *figura 1* illustra una delle mappe utilizzate nell'implementazione e nel test delle varie strategie.

Anche altre mappe sono state utilizzate, alcune con meno navi o con più posizioni note al principio, ma di esse non è tenuta traccia dato l'elevato numero.

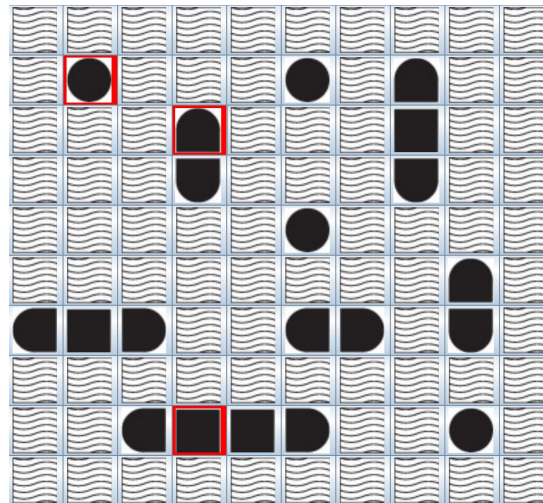


Figure 1: Tabellone di gioco con navi

# 1 Agente prima versione

## 1.1 Ordine di esecuzione - Step della strategia

- Stampa dati noti all' inizio, in caso non ve ne siano stampa "*No data available for any cell*".
- Esegue "guess" su celle note a priori e asserisce acqua nelle k-cell tutte intorno.
- Esegue azione di "guess" su celle a fianco di celle note, esse sicuramente non hanno acqua (es. prima cella sotto un top, o a destra di un left), quindi asserisce acqua tutto intorno.
- conoscendo una cella "middle" controlla se cella sopra o sotto contengono acqua, in tal caso esegue "guess" su celle immediatamente a destra e sinistra. Viceversa, se celle ai lati di un "middle" contengono acqua esegue "guess" su cella sopra e sotto.
- Se non sono possibili altre azioni viene eseguita l'operazione di terminazione.

## 1.2 Analisi strategia

I vari step precedentemente elencati illustrano a grandi linee quali regole modellano l'Agente e danno un'idea dell'ordine in cui vengono eseguite.

L'ordine di esecuzione è spesso dettato non da vincoli espliciti, ad esempio una dichiarazione di salience, ma dal fatto che alcune si attivano come conseguenza di altre.

Questa strategia è un primo passo nello sviluppo dell'Agente e nella comprensione del funzionamento del linguaggio. Essa ha come obiettivo quello di effettuare azioni di "guess" su celle che certamente hanno contenuto "boat" e le azioni di "fire" su celle che probabilmente contengono navi perchè cercate a partire da una cella con contenuto noto.

Questo primo agente racchiude tutte le regole che elaborano la sua conoscenza in un unico modulo, esso non utilizza template ausiliari per la gestione della conoscenza.

Non sono state utilizzate operazioni di "unguess" perchè le operazioni di "guess" non vengono fatte su celle che potrebbero contenere acqua.

## 1.3 Analisi risultati

Essendo una strategia iniziale ha parecchi limiti: se non si conosce il contenuto di celle a priori non può effettuare nessuna azione, ma anche se è conosciuto il contenuto di alcune celle, come nel caso in *figura 1*, il punteggio ottenuto finale è molto basso perchè le celle note a priori sono comunque poche. Per ottenere un punteggio alto bisognerebbe conoscere a priori almeno una cella per ciascuna nave, e preferibilmente una cella terminale come un "top" o un "left".

## 2 Agente seconda versione

### 2.1 Ordine di esecuzione - step della strategia

#### Modulo Agente

- Definizione di template e fatti ausiliari iniziale per tenere traccia di: navi ancora da scoprire per tipo, riga/colonna con n maggiore, controlli sullo stato della cella e tipologia di navi trovate.
- Se non vi sono celle note all'inizio stampa *"No data available for any cell"*.
- Se vi sono righe o colonne con  $n = 0$  asserisce acqua in tutte le celle in esse contenute.
- Asserisce acqua attorno a tutta la tabella di gioco per evitare problemi di sconfinamento durante le azioni di "fire" e "guess".
- Come nella sua versione precedente, l'Agente esegue operazioni di "guess" sulle celle note e su quelle ad esse vicine che hanno contenuto "boat" certo.
- Regole di gestione navi trovate permettono di individuare le navi, suddivise per tipologia, le cui celle sono state tutte "guessed" oppure "fired".
- Nel caso in cui non sia possibile far scattare altre regole in questo modulo ma vi siano ancora operazioni "fires" disponibili, l'agente passa il controllo al modulo "Valutazione-Fires".
- Se non sono possibili altre azioni viene eseguita l'operazione di terminazione.

#### Modulo Valutazione Fires

- Nel caso in cui non siano state individuate tutte le navi da almeno 3 caselle (tutti gli incrociatori e la corazzata), e nel caso in cui si conosca almeno una cella contenente l'inizio o la fine di una nave orizzontale o verticale, viene richiamato il modulo Fire-Forward.
- Nel caso in cui vi siano ancora "fires" disponibili, ma non vi siano più navi da almeno 3 celle da trovare e non sia noto il contenuto di nessuna cella "top", "bot", "left" oppure "right", viene richiamata l'operazione di "fire" sulla cella che ha maggiore probabilità di contenere una nave. La cella selezionata deve avere il numero della riga e della colonna più alto.

#### Modulo Fire-Forward

- Esegue operazioni di "fire" sulla cella situata 2 posizioni dopo una nota come "top", "bot", "left" oppure "right".

## 2.2 Analisi strategia

A differenza della strategia precedente, questa seconda versione dell'Agente divide le regole che effettuano operazioni di "guess", e le regole che si occupano della gestione della conoscenza, dalle regole che gestiscono le operazioni di "fire" in due moduli separati.

La gestione delle operazioni di "fire" è divisa a sua volta in due moduli: un primo modulo (Modulo Valutazione Fires) si occupa di controllare se vi siano ancora da trovare navi lunghe almeno 3 caselle, ed in caso affermativo passare il controllo al modulo Fire-Forward. In caso negativo, invece, effettua un'operazione di "fire" sulle celle che hanno più probabilità di contenere navi.

Come nella precedente strategia l'ordine degli step è dettato grosso modo dall'ordine in cui le regole scattano all'interno del modulo. Il controllo dal modulo Agente passa al Modulo Valutazione Fires grazie ad una regola con "salience" molto bassa, la quale viene attivata dopo tutte le altre regole nel modulo (eccetto quella che gestisce la teminazione).

L'utilizzo dei moduli ha permesso di dividere in modo netto le azioni che l'Agente esegue per dedurre la posizione delle navi, quindi le operazioni di "guess" e le corrispettive regole per la gestione della conoscenza, dalle operazioni che agiscono sul campo di battaglia, ovvero le operazioni di "fire".

Si è scelto di non implementare una strategia che sfrutti operazioni di "unguess" perchè si è preferito approfondire una strategia basata sull'effettuazione di azioni di "guess" mirate alle celle dal contenuto noto, o su celle che si possono dedurre a partire da esse.

A differenza della prima versione dell'Agente in questa seconda versione sono implementate regole che permettono il controllo sui confini, che asserendo acqua tutto attorno al campo di battaglia impediscono che un'azione di "guess" o "fire" possa essere effettuata su k-cell con indici inferiori a 0 o superiori a 9.

In questa versione vengono inoltre utilizzati i numeri che indicano la quantità di celle occupate in una riga (o colonna) per implementare operazioni di "fire" in celle con buona probabilità di contenere una nave.

### Modulo Agente

Questo modulo riprende molte delle regole già implementate nella prima versione dell'Agente. In questo modulo sono contenute le regole per la gestione delle operazioni di "guess", i template e i fatti per la gestione della conoscenza del campo di battaglia.

### Modulo Valutazione Fires

Il modulo per la valutazione delle operazioni di "fires" sulla base della conoscenza che l'Agente ha del campo di battaglia ha lo scopo di decidere se effettuare operazioni di "fire" a partire da una k-cell nota, oppure se aprire il fuoco in una cella con buona probabilità di contenere "boat".

La scelta di aggiungere questo modulo intermedio ha permesso di avere un

controllo maggiore sull'esecuzione delle regole, e ha permesso di separare nettamente le regole per la gestione dei "fire" che rivelano se la nave è di 2, 3 o 4 caselle, dalle regole che in mancanza di altre informazioni sparano nelle celle del campo di battaglia con maggiore probabilità di contenere una nave.

### Modulo Fire Forward

Questo modulo contiene solamente le regole per effettuare le operazioni di "fires" sulla seconda cella a partire da una nota. Si arriva ed eseguire le regole contenute in questo modulo solo se vi sono ancora navi da almeno 3 celle disponibili. Questa condizione permette di sparare in una cella che probabilmente avrà contenuto "boat", e consente inoltre di individuare se la nave che si sta analizzando è composta da 2 celle (caso in cui l'operazione di "fire" permetta di scoprire una k-cell water), 3 celle (se la k-cell colpita è a sua volta una terminale), oppure se si tratta della corazzata da 4 caselle.

## 2.3 Analisi risultati

Questa strategia rispetto alla precedente permette di ottenere migliori risultati ma soprattutto permette di trovare la posizione di alcune navi anche se non vi è nessuna k-cell nota a priori.

Tuttavia questa strategia presenta ancora molti limiti, e soprattutto non sfrutta a pieno le potenzialità dei numeri che indicano quante celle contengano navi in una determinata riga/colonna.

Se si considera la mappa di gioco riportata in *figura 1* per confrontare questa strategia con la precedente il punteggio ottenuto è molto maggiore.

Se viene scelto un campo di battaglia in cui per ciascuna nave è nota almeno una cella il risultato ottenuto è analogo alla strategia precedente, anche se in questa strategia è tenuta traccia di quali navi l'agente individua.

Se, invece, viene scelto un campo di battaglia con nessuna cella nota a priori rispetto alla precedente strategia il risultato ottenuto è nettamente migliore, vengono individuate infatti celle contenenti navi.

## 3 Agente terza versione

### 3.1 Ordine di esecuzione - step della strategia

#### Modulo agente

- La definizione di fatti e template ausiliari analoga alla versione precedente.
- se non vi sono celle note all'inizio stampa *"No data available for any cell"*.
- Analogamente alla versione precedente viene asserita acqua nelle righe/colonne con  $n = 0$ , controllando che non contengano celle su cui sono state eseguite

operazioni di "guess" o "fire". Viene asserita acqua nelle celle attorno alla tabella di gioco.

- Per ogni cella che è stata "guessed" viene decrementato di 1 l'n corrispondente alla colonna/riga contenenti la cella; lo scopo è facilitare le operazioni di "fire" sulle celle con n riga/colonna massimo.
- Come nella versione precedente, viene tenuta traccia delle navi trovate.
- Nel caso in cui non vi siano più operazioni di controllo sulla conoscenza da fare viene passato il controllo a modulo che gestisce le "guess".
- Nel caso in cui non sia possibile fare operazioni di "guess" e di controllo sulla conoscenza viene spostato il controllo sul modulo che gestisce le "fires".
- Se non sono possibili altre azioni viene eseguita l'operazione di terminazione.

#### **Modulo Guess**

- Esegue le operazioni di "guess" sulle celle note e quelle attigue, come implementato nell'Agente della precedente versione.

#### **Modulo Fires**

- Esegue operazione di "fire" su cella con n riga/colonna massimo, e se non possibile e esegue operazioni di "fire" sulle celle lungo riga con n massimo.

### **3.2 Analisi strategia**

Questa terza strategia nasce a partire dalle precedenti, ma divide nettamente la gestione della conoscenza dalle regole che effettuano operazioni di "fire" e di "guess".

Nel complesso le regole utilizzate in quest'ultima strategia sono le medesime della strategia precedente, ma la riorganizzazione nei differenti moduli permette di avere un maggiore controllo su quali regole vengono attivate.

Nella terza versione dell'Agente sono inoltre presenti regole che permettono di gestire gli indicatori del numero di celle con nave per ciascuna riga/colonna. Ogni volta che viene effettuata un'operazione di "guess", dato che si è scelto di non effettuare "guess" su celle che potrebbero contenere acqua, viene decrementato di 1 l'indicatore "n" per la corrispondente riga e colonna.

Al momento di effettuare l'operazione di "fire" sulla cella con maggior probabilità di avere contenuto "boat" vengono scelti gli indicatori riga/colonna maggiori, i quali, a differenza della precedente strategia, non tengono più traccia di quante celle per riga/colonna contengono celle-nave ma quante contengono celle-nave ancora da scoprire.

In questa strategia si è scelto di non implementare regole di "fire" 2 celle dopo

a partire da una nota perchè si è notato uno spreco di colpi: l'idea di aprire il fuoco nelle celle facendo riferimento agli indicatori "n" riga/colonna, piuttosto che a partire da una cella "top", "bot", "left" o "right" nota e sparare 2 celle dopo questa, si è rivelata molto più efficace.

Il fatto di effettuare azioni di "fire" nella cella con n riga/colonna maggiore garantisce nella quasi totalità dei casi di individuare celle contenenti navi.

### 3.3 Analisi risultati

Questa strategia permette di ottenere dei risultati nettamente migliori rispetto alle precedenti. Non può però essere considerata una strategia ottima perchè per alcune mappe, ad esempio quella in *figura 1*, non tutte le navi del campo di battaglia vengono individuate.

Rispetto alle versioni precedenti questo terzo Agente ha una strategia maggiormente focalizzata alla ricerca delle celle con grande probabilità di contenere navi ed è meno influenzato dalla conoscenza del contenuto delle celle a priori. Certamente i risultati migliori si ottengono conoscendo per ciascuna nave almeno una cella all'inizio del gioco, ma anche in caso in cui nessuna cella sia nota a priori molte navi vengono comunque scovate.

## 4 Conclusioni

La realizzazione di questo agente ha richiesto non pochi sforzi in partenza per apprendere al meglio il funzionamento di CLIPS in sé e delle strategie usate per il conflict resolution.

La prima parte di realizzazione si è concentrata principalmente sull'esplorazione del nuovo ambiente e del linguaggio in esso utilizzato.

La parte più complicata è stata quella di capire come e quando determinate regole diventano attivabili e, successivamente, vengono eseguite. Una volta presa una discreta dimestichezza con la definizione di regole in maniera appropriata, abbiamo potuto concentrare i nostri sforzi sulla definizione della strategia da utilizzare per poter realizzare un sistema esperto in grado di agire in maniera intelligente.

Per testare l'efficienza e il modus operandi dell'agente in tutte le sue varie versioni in continua evoluzione, abbiamo sfruttato diverse mappe con differenti posizionamenti e con una conoscenza a priori più o meno ampia. Questo ci ha permesso di comprendere i limiti delle versioni, permettendoci di andare ad agire sui punti più critici.

Sviluppare un agente, partendo da una versione più semplice fino a permettergli di essere in grado di effettuare in modo autonomo un ragionamento a partire da una base di conoscenza, ci ha permesso di apprendere il funzionamento di un sistema esperto.