

Relazione progetto IALAB

Maurizio Dominici, Alessandro Serra, Andrea Aloï

16 aprile 2014

Sommario

Relazione del progetto Intelligenza Artificiale e Laboratorio anno accademico 2012-2013. Illustrazione delle diverse strategie proposte, con una disamina dettagliata dei pregi e difetti di ognuna di esse, approfondendo anche i motivi delle scelte fatte sia in fase di progettazione che di implementazione. Vengono illustrate X strategie: ...

Indice

1	Introduzione	1
1.1	Agent	1
1.2	Moduli	1
1.3	Controllore	2
1.4	Modulo: Controllo del tempo rimanente	2
1.5	Modulo: Inform delle celle visibili	3
1.6	Modulo: Individuazione della cella target	3
1.7	Modulo: Controllo di uscita	3
1.8	Modulo: Esecuzione dell'algoritmo A*	4
1.9	Modulo: Movimento	4
2	Strategia di base	5

Capitolo 1

Introduzione

Tutte le strategie illustrate in questo documento hanno in comune la struttura. Il codice è strutturato in maniera gerarchica, in modo tale che ogni modulo si occupi di un compito particolare da svolgere, il tutto coordinato da una serie di controllori inseriti nel file `agent.clp`, fornito nello scheletro iniziale del progetto.

1.1 Agent

Partiamo illustrando il modulo agent: questo modulo, concettualmente, si occupa solo di gestire e coordinare i vari moduli a cui è assegnato un compito specifico. Agent controlla che la condizione di entrata per un modulo sia soddisfatta, quindi dà il focus al modulo in questione attendendo che il lavoro sia svolto. Quando questo modulo rilascerà il focus attraverso una pop, agent avrà a questo punto la possibilità di dare il focus a un altro modulo che svolgerà un altro sottocompito.

1.2 Moduli

Ogni modulo, come detto, ha un compito particolare (verranno illustrati in seguito i vari moduli con i rispettivi compiti). La struttura di ogni modulo è abbastanza semplice ed è così composta con le azioni elencate in ordine cronologico:

- riceve il focus dal modulo agent, il quale controlla che la condizione affinché sia svolto il compito assegnato al modulo specifico sia soddisfatta;
- svolge le operazioni delle quali ha responsabilità di esecuzione;

- asserisce una condizione per cui sia comprensibile, al modulo agent, che il compito richiesto è stato eseguito con successo;
- rilascia il focus mediante una pop.

Ci sono poi, a seconda delle varie strategie, delle varianti a questo comportamento, ma generalmente il flusso di esecuzione è quello illustrato. Alcune eccezioni possono essere la necessità di spezzare la routine di ogni turno, ad esempio quando il modulo del tempo inferisce che il tempo rimanente per dirigersi verso un gate scarseggia, ma entreremo nel dettaglio durante l'illustrazione specifica delle varie strategie. Verranno illustrati poco più avanti i moduli più nel dettaglio anche se, per una spiegazione esaustiva, il rimando è alla sezione della strategia considerata.

1.3 Controllore

Entriamo ora più nello specifico nel comportamento del controllore. Il controllore si occupa, come detto, di dare il focus al modulo corretto. Quest'ultimo viene fatto identificando una determinata routine da svolgere per ogni passo (identificato dallo slot step del template status fornito nello scheletro iniziale del progetto):

- controllo del tempo rimanente
- inform delle celle visibili
- individuazione della cella verso cui spostarsi
- controllo che la cella individuata permetta di raggiungere un'uscita (questo ai fini di non finire in vicoli ciechi)
- calcolo del percorso per raggiungere la cella individuata mediante l'algoritmo A^*
- computazione effettiva dei passi calcolati e memorizzati dall'algoritmo A^*

1.4 Modulo: Controllo del tempo rimanente

Questo modulo si occupa di controllare che il tempo rimasto a disposizione sia sufficiente. Generalmente questo significa controllare il tempo necessario a raggiungere l'uscita più vicina e confrontarlo con il tempo rimanente. Se

il tempo rimanente è sufficiente si continua nel flusso normale delle azioni asserendo la condizione di successo per il modulo agent. Se, invece, il tempo scarseggia si intraprende un'azione che può variare a seconda delle varie strategie.

1.5 Modulo: Inform delle celle visibili

Questo modulo ha la responsabilità di effettuare le inform di celle su cui non è ancora stata una inform secondo i criteri della strategia. Generalmente si è scelto di effettuare una inform base per ogni nuova cella avvistata, in quanto il costo dell'operazione di inform è irrisorio: tuttavia questo può dipendere dalle varie strategie.

1.6 Modulo: Individuazione della cella target

In questo modulo viene scelta la cella verso cui dirigersi (target). **Non in tutte le strategie viene utilizzato a ogni passo** e, i criteri con cui viene stabilita la cella target variano molto tra le varie strategie.

1.7 Modulo: Controllo di uscita

Questo modulo, che concettualmente è un'estensione del modulo precedente (1.6), si occupa di verificare che, una volta che verrà raggiunto il target trovato, ci sia la possibilità di raggiungere un gate. Questa è un'escamotage studiata al fine di non ritrovarsi in situazioni che portino l'UAV a non poter concludere il suo lavoro, con relativa penalità altissima. Nel caso in cui sia possibile raggiungere una cella di tipo gate dal target stabilito si prosegue normalmente nel flusso dei moduli, asserendo la condizione di successo. Viceversa, se non fosse possibile raggiungere alcun gate, si aggiunge un fatto che indichi la condizione di impossibilità di uscire da quella cella, in modo che non venga più presa in considerazione nella ricerca di nuovi target papabili; quindi si procede ritraendo il fatto di successo del modulo 1.6 così che il controllore (1.3), una volta effettuata la pop, dia nuovamente il focus al modulo 1.6 che si occuperà di trovare un nuovo target papabile.

1.8 Modulo: Esecuzione dell'algoritmo A*

Questo modulo, **pressoché invariato in ogni strategia**, computa semplicemente l'algoritmo A* partendo dalla posizione attuale per arrivare al target. All'interno del modulo stesso vengono prodotti ... **(COMPLETARE DESCRIZIONE DETTAGLIATA DEL MODULO).**

1.9 Modulo: Movimento

Anche quest'ultimo modulo, che in realtà è stato scritto, per questioni di economia di codice, direttamente nel modulo agent, è invariato per ogni strategia scelta, infatti si occupa solamente di computare un passo dell'UAV mediante la regola, già fornita nello scheletro iniziale, exec. Per eseguire l'azione di movimento ci si basa sui fatti di tipo PATH prodotti nel modulo 1.8, già ordinati, che indicano i singoli movimenti che l'UAV deve compiere a ogni passo.

Capitolo 2

Strategia di base

La prima strategia illustrata è quella più semplice, che useremo come metro di paragone.