Alma Mater Studiorum · Università di Bologna

Dipartimento di Informatica - Scienza e Ingegneria (DISI)
Corso di Laurea Magistrale in Informatica

Collaborative Robbies

Relazione di FSC

18 dicembre 2015

Giulio Biagini
0000705715
giulio.biagini@studio.unibo.it

Gianluca Iselli
0000705716
gianluca.iselli@studio.unibo.it

Indice

Capitolo 1

Introduzione

In questo capitolo andremo a richiamare alcuni cenni teorici di Intelligenza Artificiale che useremo per descrivere il comportamento dell'entità che abbiamo il compito di progettare, affinché possa compiere nel migliore dei modi il proprio dovere.

1.1 Entità Intelligenti

Lo scopo dell'*Intelligenza Artificiale* è quello di cercare di capire come le *entità intelligenti* funzionano ed, in particolare, quello di cercare di costruire, creare, entità intelligenti.
Negli anni sono state date varie definizioni di entità "intelligenti": alcune fanno paragoni con gli esseri umani, altre usano il principio della razionalità, alcune definiscono queste entità in base a come pensano, altre a come agiscono. Nel nostro particolare caso andremo a definire un'entità come intelligente se *agisce in modo razionale*, ovvero se "fa la cosa giusta".
Affinché **Robby**, l'entità intelligente che abbiamo il compito di progettare, possa essere definita come tale, dovrà dunque "fare la cosa giusta". Siccome questa dovrà muoversi in uno spazio cercando di pulirlo, questo significherà, ad esempio, non sbattere contro i muri, raccogliere le lattine durante il proprio passaggio o non tentare di raccogliere lattine dove queste non ci sono.

1.2 Agenti Intelligenti

Un *agente intelligente* è un'entità intelligente in grado di percepire l'ambiente tramite dei *sensori* e compiere delle azioni tramite degli *attuatori*.
Un agente ha delle *percezioni* dell'ambiente, ovvero l'insieme di tutti gli input

percettivi che provengono dai propri sensori in un dato istante.
Il comportamento di un agente intelligente è descritto matematicamente da una *funzione agente*, ovvero l'insieme di tutte le sequenze percettive e da tutte le relative azioni. L'implementazione di una funzione agente prende il nome di *programma agente*.
Il nostro primo obiettivo, dunque, sarà quello di dotare **Robby** di sensori che gli permettano di percepire l'ambiente ed attuatori che faranno sì che esso potrà muoversi e compiere determinate azioni.
I sensori di cui Robby sarà dotato gli permetteranno di percepire lo spazio attorno, ovvero, la presenza muri, lattine o se esso è vuoto. Gli attuatori gli permetteranno di muoversi verso nord, verso sud, a destra, a sinistra e di rimanere fermo nella posizione in cui si trova. Robby avrà poi speciali attuatori che faranno sì che possa raccogliere lattine.
Il programma agente che specificherà al robot quale azione compiere in base alla vista che avrà in quel momento (percezione dell'ambiente) caratterizzerà il comportamento del robot stesso.

1.3 Agenti Razionali

Lo scopo dell'Intelligenza Artificiale, però, non è semplicemente quello di andare a creare agenti intelligenti, ovvero entità in grado di percepire l'ambiente ed attuarvi azioni, ma, piuttosto, creare *agenti razionali*, ovvero entità intelligenti che "facciano la cosa giusta".
Questo significa che il programma agente per ogni sequenza percettiva produce un'azione che massimizza una determinata *misura di prestazione*: se le azioni attuate portano l'ambiente ad attraversare una sequenza di stati che potranno essere definiti "desiderabili", allora il programma agente massimizza tale misura di prestazione.
Nel caso di **Robby**, la misura di prestazione da massimizzare sarà il numero di lattine che saranno mediamente raccolte in un dato ambiente.

1.4 Agenti Reattivi Semplici

Esistono varie tipologie di programmi agente in base alle tipologie di agenti di cui questi hanno il compito di descrivere il comportamento. I più semplici di tutti sono gli *Agenti Reattivi Semplici*, ovvero agenti che basano le proprie azioni solamente sulla percezione corrente: in base a quanto letto dagli input percettivi in un dato istante, è calcolata l'azione da compiere.
Robby, che sarà in grado di "vedere" una porzione dello spazio che lo circon-

da, attuerà delle azioni che saranno guidate solamente da quanto percepito in quel dato momento. Esso sarà dunque un agente reattivo semplice.

1.5 Ambiente

L'*ambiente* nel quale gli agenti razionali si muovono possono essere di vario tipo in base alle caratteristiche che possiedono.
Lo spazio nel quale **Robby** dovrà muoversi sarà discretizzato in celle, ogni cella potrà essere occupata da un ostacolo (muro), da una lattina o potrà essere vuota. Le azioni di movimento permetteranno al robot di muoversi da una cella all'altra mentre, le lattine potranno essere raccolte solamente se il robot si troverà su una casella da esse occupata.
L'ambiente sarà *parzialmente osservabile*, ovvero solo il contenuto di alcune celle potrà essere rilevato e *stocastico*, in quanto è presente l'azione random che permetterà ai robot di poter attuare una delle azioni precedentemente elencate scelta casualmente.

1.6 Agenti in Grado di Apprendere

Esistono particolari tipologie di agenti che possono essere programmate in modo che siano *in grado di apprendere*. Un agente, infatti, può basare la scelta delle proprie azioni su conoscenze che ha dell'ambiente che sono state inserite da un programmatore: in questo caso si dice che l'agente manca di autonomia. Al contrario, un *agente autonomo* è in grado di apprendere per compensare la presenza di conoscenza parziale o erronea.
Un agente in grado di apprendere ha il vantaggio di poter operare in ambienti all'inizio sconosciuti, diventando col tempo via via più competente.
Un agente in grado di apprendere possiede oltre ad un *elemento esecutivo*, che gli permette di selezionare le azioni da compiere, anche di un *elemento di apprendimento*, responsabile del miglioramento interno, il quale usa le informazioni provenienti dall'*elemento critico* riguardo le prestazioni correnti dell'agente e determina se e come modificare l'elemento esecutivo affinché in futuro si comporti meglio ed un *generatore di problemi* il cui scopo è quello di suggerire azioni che portino ad esperienze nuove e significative dalle quali apprendere.
Nel caso di **Robby**, quello che vogliamo non è solo un agente autonomo, ma anche un agente in grado di apprendere. Per fare questo ci appoggeremo ad *Algoritmi Genetici*.

1.7 Algoritmi Genetici

Gli *Algoritmi Genetici* sono algoritmi che sono spesso usati in quelle situazioni nelle quali risulta difficile andare a progettare una determinata strategia che ci permetta di arrivare ad una soluzione.
Questi algoritmi sono difatti usati per far sì che siano loro stessi ad evolvere il sistema in modo che esso arrivi autonomamente ad una soluzione.
Uno dei campi nei quali sono maggiormente impiegati, infatti, sono i *Sistemi Complessi*.
Per far sì che **Robby** possa muoversi e pulire l'ambiente raccogliendo il maggior numero di lattine con il minor numero di mosse, non conoscendo a priori la posizione del robot stesso nella mappa, né delle lattine, non rimane che progettare un algoritmo genetico che permetta ai robot di apprendere una buona strategia.
Dapprima sono generati n individui con un *dna* casuale, ovvero, ad ogni possibile vista è associata una mossa casuale. Dopodiché si valuta ciascun robot per un numero p di passi su di una mappa nella quale le posizioni delle lattine sono generate casualmente, così come la posizione del robot. Si ripete la valutazione su m mappe per ogni robot assegnando, alla fine, un *valore di fitness* medio a ciascuna entità. Questo valore è stabilito da una *funzione di fitness* che valuta ogni mossa del robot: se questo muove contro un ostacolo viene punito con un valore negativo, se tenta di raccogliere una lattina dove essa non è presente viene altresì puntio, mentre, nel caso in cui una lattina sia raccolta viene premiato. Alla fine della fase di valutazione, i robot sono ordinati in base al valore di fitness, ed ha inizio la generazione della nuova popolazione: sono scelti due individui in base al valore di fitness o in base alla posizione nel ranking globale e viene attuato il *crossover*. Questa metodologia consiste nella scelta di un punto casuale del dna dei due genitori e, unendo rispettivamente la prima metà del dna del primo genitore con la seconda metà del dna del secondo genitore viene generato il primo figlio ed unendo la seconda metà del dna del primo genitore con la prima metà del dna del secondo genitore viene originato il secondo figlio. Infine sono presi tutti i *geni* (vista/azione) di ogni figlio e con una probabilità x sono *mutati*, cioè, alla vista è modificata in modo casuale l'azione corrispondente. Questo procedimento di generazione di una nuova generazione a partire dalla vecchia che viene poi a sua volta valutata, viene iterato g volte. È così che, dopo svariate generazioni, i robot apprendono strategie di pulizia che permettono loro di arrivare molto vicino all'obiettivo.

Capitolo 2

Obiettivo

In questo capitolo sono descritti gli obiettivi di questo lavoro. In particolare, quello che ci interessa non è tanto realizzare un sistema nel quale un'entità si muove in un ambiete e cerca di ripulirlo dalle lattine, imparando come fare grazie all'implementazione di un algoritmo genetico, ma studiare come, inserendo due entità nell'ambiente, queste si influenzino a vicenda. Lo scopo principale, dunque, è analizzare se può esservi **collaborazione** spontanea o meno fra le due entità.

2.1 Entità Invisibili - Vista a Croce

Con la prima prova che andremo a fare, tenteremo di stabilire quali sono le performance di due robot che si muovono sulla mappa senza vedersi, ovvero ignorando le mosse dell'altro robot. Sarà dunque ammesso trovarsi nella stessa cella e, nel caso in cui entrambi raccolgano una lattina posizionata in uno stesso posto, il secondo a raccogliere non riceverà un punteggio negativo. I due robot avranno vista a croce, ovvero potranno vedere la casella a nord, quella a sinistra, quella nella quale si trovano, quella a destra e quella a sud.

2.2 Entità Invisibili - Vista Quadrata

Come nel caso precedente, le due entità non saranno in grado di vedersi ma avranno una percezione maggiore della mappa nella quale si trovano: potranno infatti vedere uno spazio di 3x3 celle dove il robot occupa la posizione centrale (seconda riga e seconda colonna).
Questa prova ci permetterà di capire se potendo vedere una porzione di mappa maggiore, i robot sono avvantaggiati nella pulizia della stessa, ovvero, se

disponendo di più "risorse" raggiungono un obiettivo che massimizza maggiormente la misura di prestazione, sia essa la raccolta di un numero maggiore di lattine, sia essa la raccolta dello stesso numero di lattine ma con un numero minore di passi.

2.3 Entità Visibili - Vista a Croce

Con quest'ultima prova vedremo cosa succede dando la possibilità ai due robot di vedersi. In questo caso, sarà punita la presenza di entrambi nella stessa casella. I due robot dovranno dunque imparare a non tentare di occupare la setssa cella.
Vedremo così se i due robot impareranno a collaborare o se invece la presenza di uno sotacolerò l'altro, raggiungendo valori minori della funzione di fitness.

Capitolo 3

Risultati