

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

Daniele Baschieri
0000688992
daniele.baschieri@studio.unibo.it

Indice

1	Introduzione	5
1.1	Robby: The Soda-Can-Collecting Robot	5
1.2	Entità Intelligenti	6
1.3	Agenti Intelligenti	6
1.4	Agenti Razionali	7
1.5	Agenti Reattivi Semplici	7
1.6	Agenti in Grado di Apprendere	7
2	Obiettivi	9
2.1	Entità Invisibili - Vista a Croce	9
2.2	Entità Invisibili - Vista Quadrata	9
2.3	Entità Visibili - Vista a Croce	10
3	Progettazione	11
3.1	Algoritmi Genetici	11
4	Implementazione	13
5	Risultati	15
6	Conclusioni	17

Capitolo 1

Introduzione

In questo capitolo andremo a descrivere *Robby: The Soda-Can-Collecting Robot* [1], il progetto a cui questo lavoro si ispira, riportando alcuni cenni teorici di Intelligenza Artificiale che useremo per descrivere in maniera più formale la struttura dell'intero progetto.

1.1 Robby: The Soda-Can-Collecting Robot

Robby è un robot il cui compito consiste nel muoversi in uno spazio sporco e cercare di ripulirlo. Lo spazio può essere visto in modo astratto come una griglia suddivisa in celle. Robby possiede solamente una vista parziale dello spazio nel quale si trova: è in grado di vedere cosa è presente nella cella a nord, in quella a sinistra, nella casella in cui si trova, in quella a destra ed in quella a sud rispetto alla propria posizione. Le celle possono essere “pulite” (vuote), “sporche” (nelle quali è presente una lattina da raccogliere) oppure rappresentare un ostacolo (un muro).

Robby attua un'azione in base alla vista che ha in quel momento dell'ambiente. Se, ad esempio, vede una lattina nella cella a nord, un muro nella cella a sinistra e niente nella cella da lui occupata, in quella a destra e nella cella a sud, questo non si muoverà negli spazi vuoti, né tantomeno sbatterà contro il muro muovendosi a sinistra, ma si muoverà di un passo verso l'alto, in direzione della lattina, per poi raccoglierla. Le azioni che Robby può effettuare sono: la mossa verso nord, verso sinistra, verso destra e verso sud, può decidere di rimanere fermo, raccogliere una lattina oppure effettuare una mossa casuale scelta fra le precedenti. Tutte le mosse hanno l'effetto immaginato a parte la raccolta della lattina. Questa toglierà la lattina dalla mappa nella cella in cui il robot si trova, se presente, altrimenti lascerà l'ambiente invariato.

Inizialmente Robby possiede un dna casuale, ovvero, ad ogni vista è associata un'azione casuale. Robby è però in grado di apprendere come muoversi correttamente nell'ambiente e come raccogliere lattine. La sua evoluzione è infatti guidata da un Algoritmo Genetico.

1.2 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 creare, possa essere definita intelligente, dovrà quindi “fare la cosa giusta”. Intuitivamente, siccome il compito del robot sarà quello di muoversi in uno spazio cercando di pulirlo, “fare la cosa giusta” significherà, ad esempio, non sbattere contro i muri durante il proprio movimento, raccogliere le lattine durante il proprio passaggio e non tentare di raccogliere lattine dove queste non sono presenti.

1.3 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*.

Analizzando la definizione di agente intelligente, possiamo notare come **Robby** sia un'entità che appartenga a questa categoria: esso infatti è in grado di percepire l'ambiente tramite dei sensori che gli permettono di vedere il contenuto delle celle che lo circondano e possiede degli attuatori che gli consentono di muoversi e raccogliere lattine.

L'insieme di tutte le viste (percezioni) che Robby può avere dell'ambiente e le rispettive azioni sono descritte da una funzione agente. Il nostro compi-

to sarà quello di fornirne un'implementazione attraverso la scrittura di un programma agente.

1.4 Agenti Razionali

Il nostro obiettivo, però, non è semplicemente quello di creare agenti intelligenti, ovvero entità in grado di percepire l'ambiente e compiere azioni, ma, piuttosto, creare *agenti razionali*, ovvero entità sì intelligenti, ma che “facciano la cosa giusta”. Questo significa che il programma agente, per ogni sequenza percettiva, produce un'azione che va a massimizzare una determinata *misura di prestazione*: se le azioni attuate dall'agente portano l'ambiente ad attraversare una sequenza di stati che può essere definita “desiderabile”, allora la misura di prestazione è massimizzata.

Nel caso di **Robby**, la misura di prestazione da massimizzare sarà sia il numero di lattine che debbono essere raccolte in un dato ambiente, sia il numero di passi impiegato per raccoglierle.

1.5 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. Questi agenti, dunque, analizzano tutti gli input percettivi che provengono dai sensori in un dato istante e computano quale azione compiere.

Robby è un agente reattivo semplice in quanto la scelta dell'azione da attuare è guidata solamente dalla vista che ha in quel momento dell'ambiente.

1.6 Agenti in Grado di Apprendere

Esistono particolari tipologie di agenti che possono essere programmati in modo che siano *in grado di apprendere*.

Un agente, infatti, può calcolare la scelta delle proprie azioni basandosi su conoscenze pregresse che ha dell'ambiente, che sono ad esempio state inserite a priori 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.

Questa tipologia di agenti possiedono, oltre ad un *elemento esecutivo*, che gli permette di selezionare le azioni da compiere, anche 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. Infine, le entità in grado di apprendere possiedono 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 è un agente autonomo in grado di apprendere, ovvero un agente che non si basi su conoscenza pregressa da noi inserita. Per fare questo ci appoggeremo sugli *Algoritmi Genetici*.

Capitolo 2

Obiettivi

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 ambiente 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 stessa cella.

Vedremo così se i due robot impareranno a collaborare o se invece la presenza di uno sotacolerà l’altro, raggiungendo valori minori della funzione di fitness.

Capitolo 3

Progettazione

3.1 Algoritmi Genetici

Gli *Algoritmi Genetici* sono algoritmi spesso usati in tutte 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 punito, 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 me-

to-dologia 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 4

Implementazione

Capitolo 5

Risultati

Capitolo 6

Conclusioni

Bibliografia

- [1] Melanie Mitchell,
“*Robby, The Soda-Can-Collecting Robot*”.
[http://web.cecs.pdx.edu/~mm/ArtificialIntelligenceFall2008/
Homework/Homework6.pdf](http://web.cecs.pdx.edu/~mm/ArtificialIntelligenceFall2008/Homework/Homework6.pdf).