

Digital Pheromones in Multiagent Systems

di Stefano Montesi

1. INTRODUZIONE

Molte specie di insetti sociali gestiscono e coordinano le loro attività senza grandi ragionamenti computazionali o comunicazione diretta, essi sfruttano infatti la presenza di un ambiente comune e condiviso in cui ciascuno rilascia delle sostanze chiamate feromoni in base a cui la comunità reagisce e si comporta: questo porta ad avere una coordinazione robusta e condivisa.

Per questo motivo i suddetti meccanismi possono essere utilizzati in altri sistemi ingegneristici che hanno a che fare con la coordinazione in genere: i feromoni possono essere visti come un metodo alternativo per costruire un campo potenziale che possa guidare dei movimenti fisici coordinati. Le dinamiche stigmergiche seguite dagli insetti sono molto adatte per gestire problemi come il controllo di veicoli robotici indipendenti, problema che analizzeremo di seguito.

I campi potenziali di cui ci occuperemo sono caratterizzati dai seguenti punti:

- diversità (integrazione di informazioni provenienti da fonti e ambiti diversi);
- distribuzione (posizione ravvicinata rispetto a dove è stato generato e a dove deve essere utilizzato);
- decentralizzazione (deve essere possibile effettuare decisioni locali e non dover passare sempre per un controllo centrale);
- dinamicità (l'ambiente cambia rapidamente e continuamente).

I feromoni si caratterizzano invece per le seguenti attività:

- aggregazione (le informazioni depositate da organismi differenti e in tempi diversi vengono fuse tra loro);
- diffusione (le informazioni depositate in un determinato luogo si diffondono in modo da essere utilizzabili da organismi vicini);
- evaporazione (le informazioni vengono dimenticate col passare del tempo a meno che non vengano rinforzate nell'arco di un certo periodo di tempo).

2. CARATTERISTICHE DEL MODELLO

Una qualsiasi implementazione di questo modello, e di conseguenza anche quella che prendiamo in considerazione in questo momento, deve comprendere:

- un ambiente (che contiene il campo potenziale ed esegue aggregazione, diffusione, evaporazione);
- degli agenti “walker” (che si muovono reagendo al campo presente e lo modificano depositando ulteriori feromoni).

L'implementazione in oggetto prevede l'utilizzo di due diversi tipi di agenti:

- i place agents costituiscono l'ambiente (con relazioni di vicinato che definiscono l'adiacenza);
- i walker agents rappresentano ciascun agente walker.

Caratteristiche dei place agents:

- lista dei vicini (con conseguente identificazione della posizione di ciascun vicino);
- coordinate che ne identificano la posizione nell'ambiente (forma quadrata, semplificazione rispetto al modello originale);
- un attributo numerico per ogni tipo di feromone che ne identifica la quantità presente;
- metodi per modificare i livelli di ciascun tipo di feromone in base a diffusione ed evaporazione.

Deve essere quindi considerata la scelta del fattore di propagazione (diffusione):

- un fattore piccolo garantisce un'ottima precisione nelle zone a breve distanza dagli obiettivi ma lascia inesplorate le ampie zone lontane dagli obiettivi;
- un fattore grande garantisce invece informazioni più generiche anche lontano dalla zona degli obiettivi ma allo stesso tempo risulta meno preciso nella zona degli obiettivi.

Nel nostro caso, per questioni di semplicità, preferiamo avere informazioni che riflettano caratteristiche diverse dello stesso ambiente: computeremo prima una funzione pesata dei vari tipi di feromoni presenti in ciascun place, in modo da avere un unico campo il cui gradiente può essere seguito dai walker agent. Avremo quindi semantiche differenti e potremo scegliere in fase di configurazione come vogliamo gestire le dinamiche.

Si presenta quindi la necessità di scegliere in quali momenti aggiornare i valori dei feromoni e bisogna quindi domandarsi se può aver senso aggiornare tutti i valori nello stesso istante: riteniamo che la soluzione più chiara sia quella di separare gli aspetti dovuti a responsabili differenti (i walker, il tempo e lo spazio).

Possiamo considerare tre tipi di feromoni differenti:

- uno pseudo feromone che indica la distanza dall'obiettivo: presente all'istante zero nel punto in cui si trova l'obiettivo stesso, il livello cala con l'aumentare del raggio per via della diffusione ma esso non è soggetto ad evaporazione dal momento che l'obiettivo è fermo e la fonte è costante;
- uno pseudo feromone che indica la potenza della base: presente all'istante zero nel punto in cui si trova la base da raggiungere: il livello cala con l'aumentare del raggio per via della diffusione ma esso non è soggetto ad evaporazione dal momento che la base è fissa e con potenza costante;
- un feromone vero e proprio che viene rilasciato dai walker agents durante i propri movimenti e che è soggetto ad evaporazione e diffusione. Per comodità adotteremo un modello in cui ciascun walker distribuisce una quantità costante di feromone dividendola in parti uguali su ciascun place visitato durante il suo percorso, per fare ciò il deposito avviene al raggiungimento della base da parte del walker.

Caratteristiche dei walker agents:

- posizione corrente identificata tramite coordinate come nel caso dei place agents;
- metodi per il deposito di feromoni (qualità/quantità);
- metodi per la percezione dei valori di feromoni (place agent corrente e place agents confinanti);
- metodo per la scelta del prossimo place agent da raggiungere.

Caratteristiche dell'algoritmo di routing:

In linea di massima il movimento di ciascun walker agent si basa sulla percezione delle concentrazioni dei diversi feromoni presenti nello spazio del vicinato: un valore basso del gradiente che indica il percorso da seguire, potrebbe però essere dovuto al rumore di sistema e quindi risulta preferibile che l'agente continui nella direzione precedente oppure che esplori liberamente.

Per combinare hill climbing ed esplorazione stocastica si utilizza quindi una roulette pesata, i cui pesi sono legati ai valori dei diversi feromoni nei place agents vicini e la cui funzione di mapping determina il grado di stocasticità.

Se definiamo s_i' come la somma delle concentrazioni dei diversi tipi di feromone presenti nel place i , divisa per il numero dei tipi, allora il peso normalizzato p in quel place sarà:

$$p_i = s_i' / \sum_{j=1} s_j'$$

Di conseguenza la probabilità p_i' che un walker si muova su quel place è:

$$p_i' = e^{\beta * p_i} / \sum_{j=1} e^{\beta * p_j}$$

Il parametro beta determina il grado di stocasticità del movimento del walker:

- $\beta < 4$: probabilità di selezione sono più simili di quanto effettivamente indichino i feromoni quindi avremo un gradiente basso e sarà preferibile l'esplorazione;
- $\beta > 5$: enfatizza le differenze fra i feromoni quindi avremo un gradiente alto e sarà preferibile seguire le indicazioni date dal gradiente.

3. MAPPING DEL MODELLO ARCHITETTURALE IN JADE

3.1 Implementazione del modello sul middleware scelto

A questo punto abbiamo deciso di utilizzare come piattaforma di sviluppo uno tra i middleware distribuiti esistenti più comodo per il nostro modo di lavorare: Jade ci permette infatti lo sviluppo di applicazioni distribuite utilizzando il paradigma della programmazione ad agenti ed estendendo il linguaggio java con un insieme di servizi ad hoc.

Risulta abbastanza immediato effettuare un mapping uno-uno fra gli agenti descritti in precedenza e gli agenti presenti nel modello infrastrutturale di jade: ovviamente avremo anche in questo caso due agenti con behaviours di tipo diverso, i quali possono interagire fra loro tramite messaggi e invocazione di servizi che ciascun place agent ha registrato nel DF in fase di inizializzazione.

Nel nostro caso l'interazione avviene sempre tramite messaggi in quanto le diverse entità presenti conoscono i nomi e gli indirizzi che gli permettono di comunicare direttamente fra loro sfruttando nella maggior parte dei casi identificatori come la posizione all'interno dell'ambiente oppure il tipo di agente da contattare.

In suddetta fase verrà inoltre creata una figura geometrica composta da un numero predefinito di agenti, ognuno dei quali rappresenta un quadrato, in modo tale da formare l'ambiente di sviluppo degli esperimenti: è inoltre fondamentale inserire le giuste relazioni fra ciascun place agent in modo da riprodurre fedelmente la situazione geometrica reale.

Può aver senso creare un agente padre che chiameremo “environment” e che richiederà un parametro di costruzione: dal momento che per semplicità costruiremo un quadrato, tale parametro ci informerà sul numero di place agents presenti su ogni lato e quindi dimensionerà l'ambiente stesso. L'environment, al momento della sua creazione, provvederà quindi alla creazione della struttura di cui abbiamo parlato in precedenza passando ai vari place la dimensione dell'ambiente, grazie a tale informazione saranno i place stessi a settare i giusti vicini semplicemente ragionando sulla propria posizione.

Una volta creato l'ambiente sarà l'utente a far scatenare l'esecuzione attraverso la pressione dei quadrati che rappresentano i place: la pressione con il tasto sinistro del mouse corrisponderà alla volontà di disporre una base in quel place, viceversa se sarà effettuata con il tasto destro allora da quel place partirà la creazione e l'avvio di un walker.

Alla creazione delle basi si genererà un campo potenziale con un gradiente (visibile graficamente), il quale dovrà essere seguito dai walker durante la loro attività con l'obiettivo di raggiungere una delle basi presenti e terminare così il loro ciclo di vita.

L'algoritmo di routing viene cablato all'interno dei behaviours, in questo modo può essere immediato notare la differenza fra due walker che adottano metodi di orientamento diversi.

Per valutare l'efficienza dell'algoritmo e i diversi parametri di evoluzione dell'ambiente, consideriamo il numero dei place agent visitati dal walker prima di raggiungere una base: i motivi principali sono le ipotesi di uguale velocità fra gli agenti e la non predicibilità dello scheduling dei comportamenti dei diversi agenti.

3.2 Ulteriori considerazioni tratte dall'analisi del problema proiettato su Jade

Tutto quanto descritto finora si svolge in un'unica piattaforma e nel solo main container: nel nostro caso infatti, non ci sono motivazioni particolari che ci invitino a distribuire i place su piattaforme o container differenti, e in questo modo consolidiamo l'immagine dell'environment come un vero e proprio contenitore compatto.

Nel caso in cui si volesse invece realizzare un vero e proprio sistema distribuito, si potrebbero identificare diverse zone dello stesso ambiente dislocandole in container differenti oppure diversi ambienti realizzandoli in piattaforme separate. In quel caso però andrebbe completamente ridisegnato il modello e l'interazione fra i soggetti che vi partecipano.

Per quanto riguarda gli aspetti di mobilità, copia e spostamento degli agenti, la nostra implementazione rappresenta la situazione attraverso gli stessi concetti rivisti in maniera “virtuale”: non è il vero e proprio walker che si sposta fisicamente su un place per volta, ma sono i messaggi scambiati con i vari place a concedergli la possibilità di conoscere le informazioni relative al place corrente.

Attraverso questa metodologia ogni walker ottiene informazioni come le coordinate, il livello dei feromoni e la posizione dei place vicini solo attraverso lo scambio dei messaggi e senza avere un riferimento vero e proprio al place che sta virtualmente occupando.

Utilizzando questa tecnica possiamo sfruttare la possibilità che Jade offre in termini di oggetto del messaggio, trasferendo copie degli oggetti Java che incapsulano le nozioni necessarie, senza dover appunto trasmettere un agente per intero.

Nella gestione della parte grafica è risultata abbastanza problematica la mancata possibilità di passaggio di oggetti per riferimento: sarebbe stato molto più pulito architetturealmente se si fosse potuto accoppiare il JButton identificante il place con l'agente place stesso; questa operazione è risultata impossibile in quanto tale oggetto viene creato dall'environment che lo aggiunge al frame e non può invece passarlo al place stesso.

Per questo motivo l'implementazione della parte grafica ha di per sé reso meno lineare la struttura del progetto in quanto ogni qualvolta il place deve aggiornare la sua corrispondente parte grafica, lo può fare solamente inviando un messaggio all'environment il quale reagirà invocando metodi sul JButton corrispondente.

4. DINAMICA

4.1 Dinamica della configurazione

- 1) Lanciare un Environment tramite la RemoteAgentManagement GUI o da terminale specificando il numero di PlaceAgent presenti su ogni lato del quadrato che rappresenterà lo spazio dei movimenti.
- 2) L'environment crea i PlaceAgent necessari in base al parametro specificato: ciascuno di essi avrà un nome del tipo “PlaceAgent-i-j” dove “i” e “j” rappresentano le coordinate x,y del place all'interno dell'ambiente.
- 3) L'environment si occupa inoltre della creazione della parte grafica: una griglia di quadratini che mostrano la concentrazione di feromoni presenti in ciascun place attraverso il livello di nero dello

sfondo. L'utente ha la possibilità di scegliere il posizionamento della base cliccando con il tasto sinistro del mouse.

4) A questo punto l'environment invia un messaggio "setBase" al/i PlaceAgent corrispondente/i, quindi invia un messaggio a tutti i place (eccetto la base) specificando la posizione della/e base/i ed invitandoli a calcolare (secondo dei parametri e la loro distanza) i livelli dei propri feromoni realizzando così la pseudo-diffusione.

Da questo momento l'ambiente è stato configurato: il meccanismo precedente ha preparato lo spazio dei movimenti in modo da creare il gradiente che i walker dovranno seguire; stiamo parlando di pseudo-diffusione in quanto i feromoni relativi alla distanza e alla potenza del segnale della base non sono soggetti all'evaporazione e quindi non possono essere considerati feromoni veri e propri.

5) Una volta conclusa questa fase di assestamento del sistema, deve/devono essere creati i walker agent, i quali seguendo il gradiente devono raggiungere la base: l'operazione di creazione avviene cliccando col tasto destro del mouse sul place dal quale si vuole far partire il walker.

4.2 Dinamica dell'esecuzione del walker

Il walker agent articola i suoi comportamenti grazie ad una macchina a stati ciclica che termina al raggiungimento della base e si sviluppa nel seguente modo:

- 1) Manda un messaggio al place nel quale si trova per sapere se è in una base
- 2) Nel caso in cui non lo fosse, invia un ulteriore messaggio per conoscere gli indirizzi dei propri vicini.
- 3) Invia a ciascuno di essi un messaggio per conoscere i livelli di feromoni presenti nei vicini
- 4) Attende di ricevere tante risposte quanti messaggi ha inviato
- 5) In base all'algoritmo descritto in precedenza sceglie la prossima mossa da compiere e si sposta in un nuovo place.

Nel caso in cui il walker raggiungesse la base, invierebbe un messaggio a ciascun place visitato con il livello di feromoni di tipo walker da settare.

4.3 Dinamica dell'esecuzione del place

Il place agent incapsula diversi comportamenti che hanno il semplice compito di rispondere ai messaggi del walker informandolo sulle proprie caratteristiche.

Oltre a ciò, il place agent deve eseguire l'evaporazione del feromone di tipo walker ad intervalli di tempi regolari ed effettuare la diffusione dello stesso tipo di feromone ai propri vicini una volta ricevuto il messaggio dal walker che ha terminato il proprio percorso.

4.4 Dinamica dell'aggiornamento della grafica

Come espresso in precedenza, ogni qualvolta il place effettua dei cambiamenti riguardanti il livello di feromoni presenti al suo interno, dovrà inviare un messaggio all'environment specificando la sua posizione e il valore della somma dei diversi tipi di feromoni.

Quando l'environment riceve messaggi che richiedono l'aggiornamento grafico, li distingue in base ai template e reagisce invocando metodi diversi dei JButton.

5. ESPERIMENTI

5.1 Unica base da raggiungere, scelta dei parametri migliori

Durante la configurazione e l'esecuzione, il sistema dipende completamente da alcuni parametri sia per quanto riguarda l'algoritmo di routing sia per il livello di feromoni presenti in ciascun place.

Effettueremo quindi diversi esperimenti per identificare quali valori dei suddetti parametri ci consentono di ottimizzare il sistema, tenendo comunque conto della stocasticità presente a livello di routing.

Gli esperimenti sono stati effettuati in un ambiente di lato 30 in cui i 10 walker avevano come punto di partenza l'angolo in basso a sinistra e la base si trovava esattamente nell'angolo opposto: per questo tipo di percorso il cammino minimo è costituito da 30 mosse.

La prima parte degli esperimenti è stata svolta senza considerare il deposito di feromoni di tipo walker ed era indirizzata verso l'ottimizzazione dei seguenti parametri:

- initialDistance (livello del feromone di tipo distanza assegnato a ciascuna base)
- initialPower (livello del feromone di tipo potenza assegnato a ciascuna base)
- distanceDiffusion (fattore di diffusione del feromone di tipo distanza)
- powerDiffusion (fattore di diffusione del feromone di tipo potenza)
- beta (parametro dell'algoritmo di routing)

Siamo partiti assegnando ai parametri i seguenti valori:

initialDistance=3; initialPower=3; distanceDiffusion=0.9; powerDiffusion=0.85; beta=4.

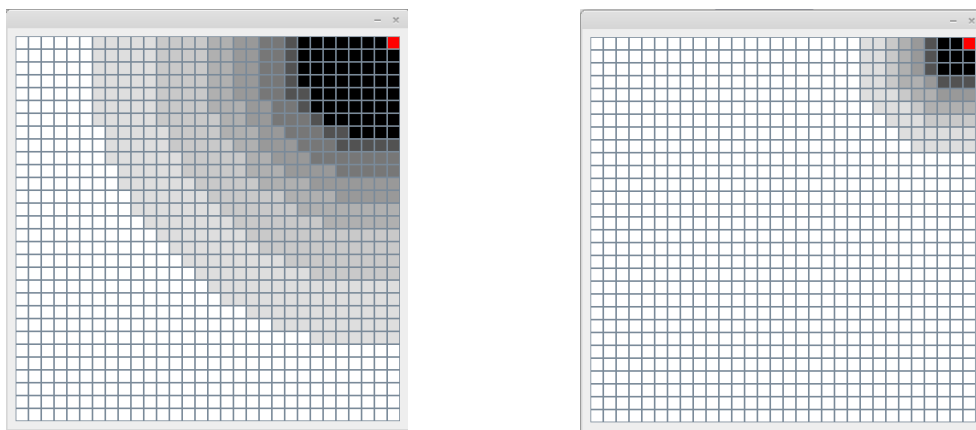
ed ottenendo pessimi risultati:

MIN:432 MAX:3061 MEDIA:1421.

Dopo diversi tentativi abbiamo trovato la configurazione ideale per i primi 4 parametri:

initialDistance=2; initialPower=2; distanceDiffusion=0.75; powerDiffusion=0.65.

In questo modo la zona con significativo livello di feromoni è considerevolmente ridotta rispetto alla situazione iniziale, come si può vedere nell'immagine,



In questo modo i walker hanno a disposizione un campo più preciso che gli permette di indirizzare il proprio cammino e allo stesso tempo le distanze non enormi ci evitano grossi problemi di esplorazione.

A questo punto entra in gioco il fattore beta: come avevamo descritto in fase di costruzione dell'algoritmo di routing, più il valore è alto e più verranno enfatizzate le differenze fra i livelli di feromoni.

A questo scopo i nostri esperimenti ci hanno portato a questi risultati (MIN, MAX, MEDIA):

- beta=4: 239, 501, 359
- beta=50: 36, 44, 40
- beta=100: 30, 37, 33

Come potete vedere, assegnando il valore 100 abbiamo addirittura raggiunto il cammino minimo e anche la media è assolutamente vicina a tale valore; per questo motivo non ci possiamo aspettare grossi miglioramenti dal settaggio degli ulteriori parametri.

Passiamo ora alla seconda parte degli esperimenti che si è svolta in maniera più dinamica rispetto alla prima parte e per tale motivo non può essere schematizzata come la precedente.

Come è stato spiegato in precedenza, i walker depositano dei feromoni nei place appartenenti al proprio cammino, in tal modo eventuali walker successivi avranno ulteriori informazioni sulla strada da intraprendere.

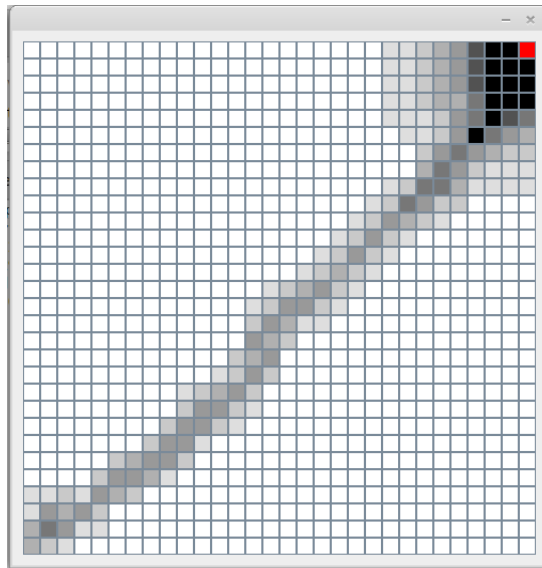
In questo caso i fattori da configurare sono i seguenti:

- deltaUpdate (ogni quanti millisecondi eseguire l'evaporazione);
- evaporationFactor (di quanto diminuire il livello di feromoni ad ogni esecuzione dell'evaporazione);
- pheroAmount (la quantità totale di feromoni da distribuire tra i place attraversati).

Abbiamo scelto di assegnare al primo parametro lo stesso valore di quel periodo di tempo che intercorre tra ogni mossa di ciascun walker in modo da bilanciare i due eventi, inoltre si è scelto un valore abbastanza prossimo ad 1 per quanto riguarda il secondo parametro per consentire di non perdere immediatamente le informazioni riguardanti i walker precedenti.

Relativamente all'ultimo parametro, c'è la necessità di dimensionarlo in base al numero di mosse che si ritiene saranno necessarie per raggiungere l'obiettivo, nel nostro caso gli è stato dato valore 20 in considerazione del fatto che saranno necessarie 30/40 mosse: così il livello di deposito sarà tale da non sovrastare le informazioni provenienti dalle basi.

Come previsto dalle considerazioni fatte poco fa, non abbiamo notato grandi miglioramenti rispetto al caso senza feromoni di tipo walker, ma addirittura si sono verificati degli episodi in cui il walker indugiava in una zona che era stata maggiormente battuta dal walker precedente.



6. RIFERIMENTI

Le attività svolte in questo report sono liberamente reinterpretate partendo dal seguente articolo:

H. Van Dyke Parunak, Sven Brueckner, John Sauter. [Digital Pheromone Mechanisms for Coordination of Unmanned Vehicles](#). 1st International Joint Conference on Autonomous Agents and Multiagent systems, 15-19 July 2002

Per tutto quello che riguarda Jade e il suo utilizzo, la pagina di riferimento è la seguente:

<http://jade.tilab.com/>