

ALMA MATER STUDIORUM
UNIVERSITÀ DEGLI STUDI DI BOLOGNA

II facoltà di Ingegneria - Cesena
Ingegneria Informatica

Self-Organisation e Emergenza in MAS

Sistemi Multi-Agente L-S

Matteo Desanti
0000341761

prof. Andrea Omicini

ANNO ACCADEMICO 2009/10

Indice

Introduzione	ii
1 Self-Organisation	1
1.1 Definizioni e Esempi	1
2 Emergenza	5
2.1 Definizione	5
2.1.1 Requisiti	6
2.2 Esempi di emergenza	6
3 Implementazione in MAS	8
3.1 Perché utilizzare i MAS?	8
3.2 Possibili approcci per i SO-MAS	9
3.2.1 Meccanismi basati su interazioni dirette	9
3.2.2 Meccanismi basati su stigmergia	10
3.2.3 Meccanismi basati su rinforzo	10
3.2.4 Meccanismi basati su cooperazione	11
4 Caso studio: Collective Sort in spazi di tuple	12
4.1 Definizione del problema	12
4.2 Modello della soluzione	13
4.2.1 Il metamodello Agenti & Artefatti	13
4.3 Progettazione degli agenti ambientali	15
4.3.1 Strategie presenti in natura	15
4.3.2 Primo modello per gli agenti ambientali	16
4.3.3 Prime simulazioni	17
4.3.4 Tuning del modello	18
5 Conclusioni	22

Introduzione

Molti sistemi naturali mostrano struttura e organizzazione a un macro livello. Esempi di tali sistemi includono le galassie, i sistemi planetari, le cellule e le società (intese come aggregazione di individui omogenei). La scienza tradizionale cerca di ricondurre tutto ciò, tramite l'approccio del riduzionismo, al comportamento dei micro componenti della materia che seguono le deterministiche leggi della natura. Questo modello lascia poco spazio per la spiegazione dell'emergenza spontanea di auto-organizzazione che si osserva in una moltitudine di sistemi inclusa la vita stessa. Nonostante la visione riduzionista sia molto utile nell'osservare le intricate relazioni tra semplicità e complessità, risulta chiaro come questo punto di vista sia incompleto per la spiegazioni di alcuni fenomeni presenti in sistemi complessi.

I sistemi naturali auto-organizzanti funzionano senza la necessità di controllo centralizzato e operano, principalmente, sulla base di interazioni locali. La particolarità principale di questi sistemi è la loro capacità di produrre spontaneamente, senza quindi un controllo esterno, una nuova organizzazione in caso di cambiamenti e perturbazioni nell'ambiente in cui sono situati. Per tale motivo ne consegue che i sistemi auto-organizzanti siano particolarmente robusti poiché la loro capacità adattativa ne garantisce la sopravvivenza. Spesso l'auto-organizzazione è accompagnata e accoppiata alla presenza di un *comportamento emergente*, nel senso che nonostante le singole parti del sistema svolgano semplici compiti, come insieme questi danno luogo a un processo complesso che emerge dalle interazioni locali dei vari componenti a livello più basso.

La sempre crescente evoluzione delle tecnologie, in particolare nell'ambito di Internet e delle comunicazioni a lunga distanza, ha creato nuove possibilità per le applicazioni software (dal mobile computing al calcolo pervasivo per citarne alcuni) che necessitano sempre più di operare in ambienti dinamici ed eterogenei sottoposti spesso a repentini cambiamenti senza la possibilità di avere una conoscenza completa delle future condizioni di contorno. Ciò impone che tali software siano flessibili, reattivi e robusti per permettere l'adattamento alle varie circostanze. È un'idea diffusa che sistemi multi-agente (MAS) coordinati da meccanismi di auto-organizzazione e emergenza ispirati ai sistemi

naturali siano un buon meccanismo per la progettazione di tali softwares.

Lo sviluppo di sistemi auto-organizzanti (self-organising system o SOS) stona purtroppo con le classiche metodologie ingegneristiche poiché basato su principi differenti. Normalmente i sistemi vengono progettati ingegneristicamente come composizione di singole parti più piccole e il cui complesso fornisce risultati predicibili. Al contrario, i SOS mostrano dinamiche non lineari che difficilmente possono essere riprodotte in un modello deterministico e inoltre si pone il problema di come progettare i singoli componenti in modo tale che il loro comportamento complessivo dia luogo a quello globale voluto.

Questo approfondimento si pone l'obiettivo di studiare le caratteristiche dei SOS e di presentare dei possibili approcci per lo sviluppo di questi ultimi sfruttando la tecnologia dei sistemi multi-agente. Nel capitolo 1 e 2 vengono rispettivamente introdotti e definiti i concetti di auto-organizzazione e emergenza, il capitolo 3 riporta i vari approcci con cui applicare queste ultime ai MAS, nel capitolo 4 si riporta un caso studio di SO-MAS e il capitolo 5 contiene le conclusioni.

Capitolo 1

Self-Organisation

1.1 Definizioni e Esempi

Il termine auto-organizzazione si ritiene essere stato introdotto originariamente nel 1947 dallo psichiatra e ingegnere inglese William Ross Ashby¹, considerato uno dei pionieri della cibernetica, il quale propose quello che chiamò “il principio dell’auto-organizzazione”. Secondo tale principio

“Un sistema dinamico, indipendentemente dal suo tipo o composizione, tende sempre a evolvere verso uno stato di equilibrio comunemente chiamato attrattore. Questo riduce l’incertezza che abbiamo sullo stato del sistema, di conseguenza l’entropia statistica, il che è equivalente all’auto-organizzazione.”

Un attrattore in generale è una regione dello spazio che “attira a sè” tutti i punti vicini con il passare del tempo e sono, nello studio dei sistemi complessi, di vitale importanza poiché catturano le dinamiche dei loro comportamenti a lungo termine. Tipicamente un attrattore può essere un punto, un percorso regolare, una complessa sequenza di stati o addirittura una sequenza infinita (detto in questo caso “strange attractor”).

Nel significato più ampio e generale del termine ci si riferisce all’auto-organizzazione di un sistema intendendo un processo interno che porta a un crescente livello di organizzazione intesa come relazione delle singole parti componenti e delle loro interazioni. Questo aumento deve essere però di tipo autonomo, ovvero guidato da forze interne e non centralizzate o meglio distribuite tra i singoli componenti elementari.

Per quanto detto l’auto-organizzazione deve rispettare i seguenti criteri, ciascuno non indipendente dall’altro:

¹Ashby, R., “Principles of the self-organizing dynamic system”, Journal of General Psychology, 1947.

- L'evoluzione di un sistema in una struttura spaziale/temporale organizzata deve avvenire senza l'intervento esterno
- Il cambiamento è sempre autonomo in ogni ristretta fase spaziale
- Lo sviluppo strutturale legato alle variabili indipendenti del sistema dipende solamente dall'influenza di regole locali

Queste caratteristiche fanno in modo che i SOS possiedano di conseguenza importanti proprietà quali la robustezza, fault tolerance e adattatività alle perturbazioni.

Esempi di diversi tipi di auto-organizzazione in natura si hanno in numerosi campi della scienza: dalla chimica alla biologia, dalla fisica alla economia, alcuni dei quali si citano nel proseguo.

Nel 1959 Grassé propose la cosiddetta teoria della *stigmergia* in seguito ai suoi studi sul comportamento degli insetti, termiti in particolare, il cui significato è “lavoro guidato mediante stimoli”. Tale principio definisce che non si necessita di interazioni dirette per permettere la coordinazione di un gruppo ma bastano, per esempio, interazioni indirette mediate, per esempio, attraverso l'ambiente circostante tramite il deposito di informazioni senza un controllo centralizzato. Nel caso delle termiti e delle formiche e di altri insetti sociali l'informazione è costituita dal deposito di una sostanza chimica detta *feromone* ed utilizzata per esempio nella costruzione dei termitai (figura 1.1) o nella ricerca del cibo. Tutto ciò dà origine alle cosiddette “Swarm Intelligences”, dove le interazioni di singoli individui dotati di ridotte capacità e senza una conoscenza dello stato globale, riesce tramite meccanismi di feedback positivi (il rilascio di feromone) e negativi (l'evaporazione dello stesso con il passare del tempo) a organizzare un comportamento di gruppo complesso e produttivo.

Altri esempi di comportamenti collettivi di animali a cui ci si riferisce come auto-organizzazione sono rappresentati dai banchi di pesci e gli stormi di uccelli. Seguendo alcune semplici regole, del tipo restare vicino a un pesce (o uccello) simile ma non troppo, e allontanarsi da altri pesci (o uccelli) diversi, sono così in grado di evitare collettivamente i predatori (figura 1.2).

Dal punto di vista fisico, il concetto di auto-organizzazione risulta strettamente legato a quello di entropia come riportato nei lavori del Premio Nobel Ilya Prigogine. Infatti secondo il fisico belga i SOS generano continuamente entropia, ma tale entropia viene attivamente dissipata al di fuori dello stesso. Ciò comporta un aumento dell'organizzazione interna al prezzo di quello dell'ambiente circostante. L'idea di base é che i sistemi aperti diminuiscano la propria entropia quando una energia esterna viene applicata al sistema, in modo tale che la materia riorganizzi sè stessa sotto questa pressione esterna per raggiungere uno stato in cui l'entropia risulta diminuita. Prigogine rilevò anche



Figura 1.1: I termitai sono un esempio di auto-organizzazione tramite stigmergia.



Figura 1.2: Banco di pesci in nuoto coordinato.

che tale auto-organizzazione tipicamente prenda luogo in sistemi non lineari, i quali sono lontani dal loro equilibrio termodinamico. Ad un primo sguardo l'approccio di Grassé e di Prigogine sembrano in disaccordo poiché nel primo l'auto-organizzazione risulta da un comportamento che occorre dall'interno del sistema (le formiche o termiti stesse), nel secondo caso è invece risultato di una energia applicata dall'esterno del sistema. Recenti studi [2] dimostrano che la visione stigmergica di Grassé in realtà sottostà alle formulazioni di Prigogine e non contraddice la seconda legge della termodinamica poiché a un'incremento dell'organizzazione a livello globale corrisponde un aumento dell'entropia a livello microscopico.

In ambito di studi biologici il concetto di auto-organizzazione è stato affiancato a quello più ampio di *autopoiesi*² o auto-creazione. Un sistema autopoietico può essere rappresentato come una rete di processi di creazione, trasformazione e distruzione di componenti che, interagendo fra loro, sostengono e rigenerano in continuazione lo stesso sistema. Le sue caratteristiche sono pertanto (1) di mantenere la propria organizzazione interna in seguito a perturbazioni ambientali e cambi strutturali e (2) rigenera i propri componenti durante le proprie operazioni. Mentre la prima condizione è una proprietà generale dei SOS, la seconda è un ben più forte vincolo (identificato appunto dal termine autopoiesi) che non li caratterizza specificatamente, rendendo i sistemi autopoietici un sottoinsieme dei SOS di cui era giusto citare l'esistenza.

²Maturana, H. and Varela, F., "Autopoiesis and Cognition: The Realization of the Living," 1980

Capitolo 2

Emergenza

2.1 Definizione

Normalmente ci si riferisce a un comportamento emergente o proprietà emergente quando un numero di entità semplici (agenti), operanti in un ambiente, danno origine a comportamenti più complessi in quanto collettività. La proprietà stessa non è predicibile e non ha precedenti, inoltre i comportamenti complessi non possono essere facilmente riconosciuti o dedotti dal comportamento delle entità del livello più basso. Il concetto di emergenza non è nuovo alla scienza e lo stesso termine si ritrova in antichi scritti greci del periodo socratico con cui ci si riferiva alla nozione “il tutto è più della somma delle parti” che nella mentalità moderna è invece una particolare definizione di *olismo*. I due concetti in realtà si discostano l’uno dall’altro poiché nonostante sia vero che un fenomeno emergente non può essere ridotto al comportamento dei suoi componenti individuali, l’emergenza è un concetto più vasto.

Varie sono le definizioni associate al concetto di emergenza. Per esempio, Daniel Hills afferma “A volte un sistema con diversi componenti semplici può esibire un comportamento dell’insieme che sembra più organizzato del comportamento delle singole parti¹”. Già da questa prima definizione si nota subito come il concetto di emergenza risulti legato a quello di auto-organizzazione. A sostegno di tale idea si riporta la definizione, più completa e recente di Goldstein:

“L’emergenza si riferisce al presentarsi di strutture, patterns e proprietà nuove e coerenti durante il processo di auto-organizzazione in sistemi complessi. Fenomeni emergenti occorrono a un macro

¹Hillis, W. D., “Intelligence as an emergent behavior: or, the songs of Eden”, in The Artificial Intelligence Debate: False Starts, Real Foundations, 1989

livello, in contrasto con i componenti e processi di micro livello da cui essi si generano.²”

Una relazione tra emergenza e auto-organizzazione è riportata anche in [4] in cui si afferma che “la più positiva manifestazione di emergenza è l’auto-organizzazione, dove un sistema allinea sè stesso a un problema e si auto-mantiene, anche quando l’ambiente circostante si modifica”. In sostanza si sostiene che l’auto-organizzazione sia uno specifico e ottimo comportamento emergente poiché aumenta la fitness del sistema in cui si presenta. A quanto detto occorre aggiungere che l’emergenza è, in generale, proporzionale alla complessità del sistema. Queste proprietà inattese possono apparire, scomparire e modificarsi in relazione al variare della complessità del sistema stesso. Per esempio due formiche non esibiranno il comportamento emergente trovato invece in un grande formicaio, ma una volta che una massa critica viene raggiunta, il comportamento globale emerge e viene mantenuto anche all’aumento delle dimensioni della colonia. Questa particolarità suggerisce, come detto, che l’auto-organizzazione sia propriamente una forma specifica di emergenza.

2.1.1 Requisiti

Alla luce di quanto detto l’emergenza si può mostrare in sistemi con i seguenti requisiti:

- **Non linearità** - il sistema deve essere caratterizzato da interazioni non lineari identificabili di solito con meccanismi di feedback.
- **Auto-organizzazione** - la capacità di autoregolarsi e adattare il proprio comportamento.
- **Lontano dall’equilibrio** - sistemi dinamici soggetti a mutamento e perturbazioni.

2.2 Esempi di emergenza

Come quanto visto precedentemente per i sistemi auto-organizzanti, la natura è un vasto bacino da cui si possono trarre vari esempi di fenomeni emergenti.

Il primo esempio è nuovamente riferito agli insetti sociali. Le formiche operaie sono deputate all’esplorazione dell’ambiente circostante alla ricerca del cibo. Quando una sorgente di cibo viene trovata ritornano al formicaio

²Goldstein, J., “Emergence as a construct: History and issues”, 1999.

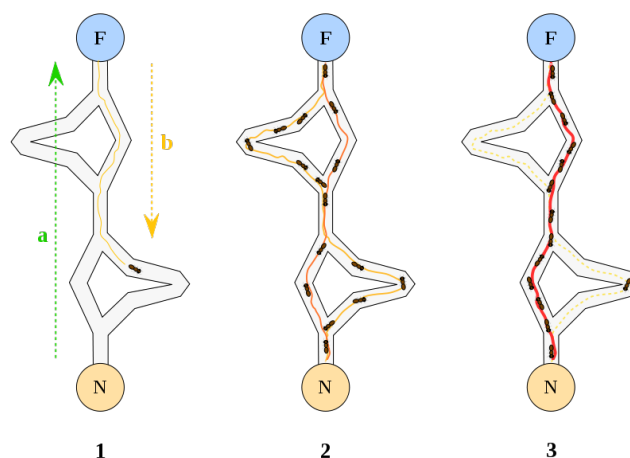


Figura 2.1: L'emergenza del percorso minimo tra tana e sorgente di cibo in seguito all'interazione stigmergica delle formiche.

lasciando, come già citato al capitolo 1, una traccia di feromone al loro passaggio. Il rinforzo di feromone dovuto al passaggio di diverse formiche dallo stesso percorso e contemporaneamente l'evaporazione dello stesso dai percorsi meno battuti fa sì che il percorso preferenziale che emerge dalle interazioni stigmergiche tra le formiche per raggiungere la fonte di cibo sia anche quello più corto possibile (vedi figura 2.1). Chiaramente questo fenomeno ha solo significato per un osservatore esterno al sistema e risulta assolutamente oscuro alle formiche.

L'apparizione di coscienza è invece un esempio di fenomeno emergente negli umani. La coscienza è vista come una proprietà del cervello a un livello globale e il cervello a sua volta è realizzato da un intricato e complesso sistema di neuroni che interagiscono tra loro. Oggigiorno la scienza non è in grado di spiegare l'apparizione della coscienza tramite la sola osservazione dei neuroni e delle loro interazioni.

Risulta interessante sottolineare come, nonostante ci si riferisca spesso ad essi con accezione positiva, i comportamenti emergenti in sistemi complessi non sono sempre di tipo benefico come in alcuni altri esempi riportati in [4].

Capitolo 3

Implementazione in MAS

3.1 Perché utilizzare i MAS?

Volendo modellare applicazioni e sistemi software che presentino caratteristiche complesse quali l'auto-organizzazione e l'emergenza di particolari comportamenti globali risulta chiaro fin da subito come la struttura su cui tali sistemi si devono basare non può essere quella classica della OOP. Il fatto stesso che si parli di sistemi fa subito intendere che il problema può essere affrontato solo mettendo in gioco una struttura composta da varie entità interagenti tra loro. Chiaramente i singoli componenti di tali sistemi devono possedere caratteristiche specifiche che permettano il generarsi dei comportamenti ricercati, tali proprietà, per citarne alcune, dovrebbero essere *situatedness*, mobilità e capacità di percepire e interagire altre entità e ambiente circostante. Se ancora un quadro del genere non fa propendere per una tecnologia implementativa particolare vi è una ultima proprietà che fa pendere la bilancia per una scelta obbligatoria, ovvero la necessità dell'*autonomia* delle singole parti del sistema. Alla luce di questo risulta praticamente obbligata la scelta della tecnologia ad agenti per la realizzazione di tali componenti, dove per agenti si intende, secondo la definizione di Maes:

“sistemi computazionali che popolano degli ambienti complessi e dinamici, percepiscono e agiscono automaticamente in tali ambienti, e facendo ciò realizzano un insieme di obiettivi o processi per i quali sono stati progettati¹”.

La tecnologia dei sistemi multi-agente (MAS appunto) risulta pertanto la più indicata e appropriata poiché adatta a riprodurre le strutture caratteristi-

¹Maes, P., “Artificial Life Meets Entertainment: Lifelike Autonomous Agents”, Communications of the ACM: Special Issue on New Horizons of Commercial and Industrial AI, 1995.

che dei sistemi auto-organizzanti come collezione di attori autonomi attraverso le cui interazioni e comunicazioni sociali far sì che si generi un comportamento globale emergente.

3.2 Possibili approcci per i SO-MAS

Studi sull'auto-organizzazione e l'emergenza si focalizzano su approcci ispirati alla natura, che a loro volta possono essere separati in approccio di ispirazione biologica e sociale, e non. I meccanismi sperimentati che portano ad auto-organizzazione e, spesso allo stesso tempo, a fenomeni emergenti sono diversi e per ciascuno di essi si può identificare un particolare approccio che sfrutta tale meccanismo.

I vari approcci possono essere indicativamente suddivisi in quattro classi e sono basati sui seguenti meccanismi:

- interazioni dirette tra agenti;
- interazioni indirette tra agenti e uso di stigmergia;
- rinforzo del comportamento degli agenti;
- comportamento cooperativo di agenti individuali.

Nel proseguo verranno presentati i vari approcci sopra indicati descrivendone le principali caratteristiche.

3.2.1 Meccanismi basati su interazioni dirette

Questa tipologia di meccanismi consiste nell'utilizzo di pochi principi base di interazione, tipo *localizzazione e broadcast*, accoppiate a interazioni e computazioni locali eseguite dagli agenti per raggiungere un coerente stato globale. Studi di algoritmi di questo tipo possono essere trovati nei lavori di Zambonelli². Tali algoritmi si differiscono dai tradizionali algoritmi distribuiti poiché sono focalizzati sull'assicurare l'eventuale convergenza e il mantenimento di uno stato desiderabile nonostante le perturbazioni dell'ambiente del tipo cambiamenti nella struttura.

Tipici esempi di tali meccanismi sono quelli applicati alle aree di auto-assemblaggio e auto-localizzazione distribuita dove è richiesta la formazione di

²F. Zambonelli, M.-P. Gleizes, M. Mamei, and R. Tolksdorf. "Spray computers: frontiers of self-organisation for pervasive computing". Second International Workshop on Theory and Practice of Open Computational Systems, 2004

patterns spaziali regolari per oggetti/entità mobili. Questi meccanismi si focalizzano sul cambiamento dell'aspetto strutturale dell'organizzazione degli agenti, tipo il posizionamento topologico degli stessi e delle linee di comunicazione tra di essi.

3.2.2 Meccanismi basati su stigmergia

I meccanismi di auto-organizzazione basati su stigmergia puntano a raggiungere comportamenti complessi di sistema come risultato di *interazioni indirette* tra agenti. Tali interazioni vengono realizzate tramite modifiche dell'ambiente facendo in modo che il comportamento degli agenti si modifichi di conseguenza per realizzare il comportamento globale obiettivo. Approcci di questo tipo sono stati proposti e utilizzati per diverse tipologie di applicazioni; per citarne alcune sono stati applicati in controllo manifatturiero, gestione della sicurezza di reti di computers, coordinazione di veicoli senza equipaggio.

Questa tipologia di meccanismi viene solitamente valutata e testata tramite un'esperienza, per esempio con successive fasi di simulazione e prototipazione. Queste fasi vengono di solito ripetute iterativamente per permettere l'ottimale calibrazione delle variabili in gioco basandosi sui risultati ottenuti. In tali casi, per via della non linearità e complessità dei fenomeni in gioco non è solitamente possibile avere controllo diretto sul comportamento del sistema né può essere provato che il comportamento desiderato possa essere raggiunto. Ciò che si può ottenere è una certa confidenza statistica che il sistema converga al desiderato comportamento coordinato globale tramite sperimentazione.

3.2.3 Meccanismi basati su rinforzo

Alcuni approcci si basano sulla capacità degli agenti di modificare dinamicamente il loro comportamento sulla base di qualche tipo di rinforzo. Tali meccanismi di auto-organizzazione si basano su questi principi base: ricompense favoriscono il comportamento degli agenti mentre punizioni lo scoraggiano. Ciò comporta che un agente individuale può adattare le sue capacità e si possono, per esempio, osservare delle specializzazioni di ruoli quindi l'auto-organizzazione si basa sulla capacità di adattamento comportamentale degli agenti inseriti in particolari strutture. In questi approcci, gli agenti dinamicamente selezionano un nuovo comportamento (o azione) basandosi sul calcolo di un valore di probabilità che dipende dallo stato attuale dell'agente e la percezione che questi ha dello stato dell'ambiente, tutto mediato dalle precedenti esperienze delle decisioni prese. Normalmente in questi contesti si fa uso di un *behaviour graph* per definire i vari ruoli che l'agente può rivestire i legami che indicano le condizioni sotto le quali si può passare da un ruolo a un altro.

La selezione di un ruolo prende piede a runtime basandosi su parametri che rappresentano proprietà degli agenti e dell'ambiente percepito i quali possono cambiare dinamicamente durante l'esecuzione.

3.2.4 Meccanismi basati su cooperazione

In questa tipologia di approcci l'auto-organizzazione viene raggiunta tramite il comportamento cooperativo degli agenti come proposto nella teoria AMAS³. Ogni agente possiede la abilità di organizzarsi, per esempio la capacità di modificare le sue interazioni locali con altri agenti e l'ambiente in funzione della sua conoscenza, della sua rappresentazione delle altre entità e dei task che deve assolvere. Capacità cooperative non intende che gli agenti siano sempre di aiuto o altruistici ma sono comunque in grado di riconoscere situazione non cooperative (*Non Cooperative Situations o NCS*), che corrispondono alle eccezioni trovate nei programmi classici, e possono gestirle. La gestione locale degli NCS massimizza la flessibilità e le capacità di adattamento del sistema a situazioni inaspettate che possono occorrere in seguito al dinamismo delle interazioni tra agenti o dell'ambiente.

³si veda in proposito [3]

Capitolo 4

Caso studio: Collective Sort in spazi di tuple

4.1 Definizione del problema

Il problema studiato in [1] è quello del cosiddetto *collective sorting*, sulla falsa riga del classico problema della robotica detto *segregation* dove un gruppo di robot esplora l'ambiente con l'obiettivo di trovare, raccogliere e separare una serie di oggetti di classi diverse. Soluzioni a queste tipologie di problemi sono spesso ricercate in natura, per esempio, la *segregation* è riscontrabile nel comportamento di insetti sociali (quali le formiche) durante l'organizzazione del cibo.

Il modello studiato in questo caso di studio intende applicare tale principio al problema del *collective sorting* in un ambiente realizzato da un insieme di N spazi di tuple e una suddivisione statica di tuple in N tipologie diverse, il servizio di ordinamento deve fare in modo che le tuple vengano spostate da uno spazio all'altro fino allo stato di pieno ordinamento in cui ogni spazio contiene solo tuple di un certo tipo. Il servizio di ordinamento messo in atto deve essere dinamico e continuo, indipendente dalle modifiche degli spazi di tuple perpetuate tramite inserimento o rimozione di tuple da parte degli agenti che li utilizzano. Non solo, lo spazio di tuple che eventualmente aggredgerà un certo tipo di tuple non deve essere *deciso a priori*, ma dipenderà probabilisticamente da come le tuple verranno spostate da uno spazio all'altro ovvero con uno stile auto-organizzante. Pertanto l'ordinamento sarà la proprietà emergente ricercata nonostante le interazione esterne, da ciò risulta chiaro che maggiori saranno gli agenti che alterano la configurazione delle tuple maggiori dovranno essere le risorse adibite al servizio di sorting per poter raggiungere convergenza.

4.2 Modello della soluzione

Il problema principale che ci si pone è come e dove iniettare i meccanismi di auto-organizzazione. L'astrazione di agente è certamente, come visto al capitolo precedente, un buon meccanismo per incapsulare i micro comportamenti che portano a pattern auto-organizzanti a macro livello, ma in alcuni casi i soli agenti falliscono nel catturare quei comportamenti che sono implicitamente distribuiti e non goal-oriented, ma essenziali in alcuni SOS. Un esempio classico in questo contesto è il processo di evaporazione del feromone che è componente fondamentale dei sistemi basati su stigmergia. Pertanto esistono una serie di problemi che sono modellati al meglio come parte dell'ambiente invece che essere espressi in termini di agenti come d'altra parte è il problema in analisi.

4.2.1 Il metamodello Agenti & Artefatti

Quanto visto al punto precedente promuove una politica di distribuzione delle funzionalità tra componenti attivi (agenti) e l'ambiente. Questo permette più facilmente il mapping dei comportamenti non goal-oriented al di fuori degli agenti riducendo la loro complessità. Ciò comunque comporta il modellamento di un ambiente con particolari requisiti, che come visto nei primi capitoli, deve offrire determinati requisiti quali:

1. supporto di interazioni indirette tra i componenti del sistema;
2. supporto al concetto di località;
3. la località deve influire sulle interazioni.

A ciò si aggiunge la necessità, in alcuni casi, di un ambiente con funzionalità attive, ovvero la presenza di processi attivi nell'ambiente stesso come può per esempio essere la funzione di evaporazione del feromone per simulare efficacemente sistemi che imitano quelli naturali. Pertanto nel caso dell'applicazione in esame, all'ambiente che normalmente consiste di un contesto che fornisce servizi di comunicazione e accesso alle risorse fisiche occorre aggiungere ulteriore profondità in modo tale da permettere di progettare e nascondere dietro a esso parte della complessità del sistema che i servizi ambientali devono offrire, liberando gli agenti da specifiche responsabilità. A tale scopo viene utilizzato il metamodello Agenti & Artefatti¹ in cui un MAS viene modellato tramite la composizione di due astrazioni fondamentali: agenti e artefatti. Gli agenti come entità proattive incapsulanti il controllo e di tipologia task/goal-driven e

¹Ricci, A., Viroli, M. and Omicini, A. "Programming MAS with artifacts", in Programming Multi-agent Systems, 2006

gli artefatti come entità passive e reattive che forniscono funzionalità e servizi sfruttabili dagli agenti attraverso delle interfacce (figura 4.1).

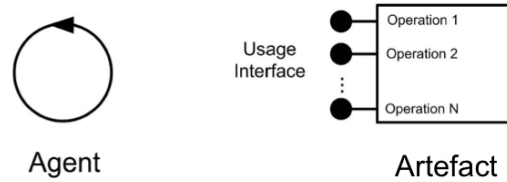


Figura 4.1: I componenti del metamodello Agenti & Artefatti.

Gli artefatti pertanto mediano le interazioni degli agenti e permettono la coordinazione di attività sociali. Purtroppo per inserire funzionalità di auto-organizzazione questa sola struttura non è sufficiente. Occorre pertanto inserire ulteriori entità a quella appena viste e a questo scopo si fa affidamento ai cosiddetti *agenti ambientali*: agenti responsabili di gestire gli artefatti in modo tale da perseguire la proprietà globale ricercata. Ovviamente gli agenti ambientali si differiscono da quelli normali poichè gestiscono gli artefatti in maniera diversa (e quindi tramite una interfaccia diversa) da quella pubblica usata dagli agenti standard detta management interface dando origine a un più complesso pattern architetturale riportato in figura 4.2.

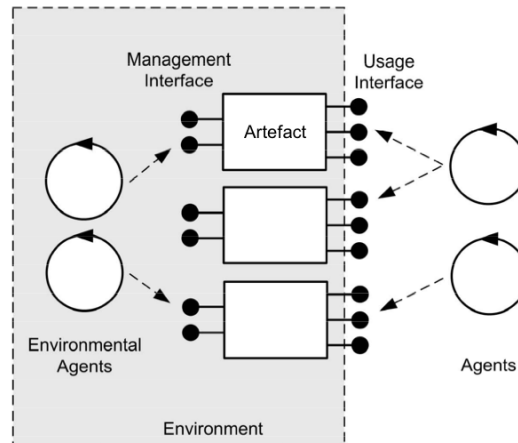


Figura 4.2: Pattern architetturale che prevede la presenza di agenti ambientali per introdurre funzionalità auto-organizzanti.

Gli agenti ambientali, essendo in qualche modo incapsulati nell'ambiente MAS, non alterano la visione che gli agenti hanno del sistema che ancora percepiscono l'ambiente popolato solo da artefatti poichè risultano completamente

trasparenti ad essi. Alla luce di ciò risulta facile il mapping tra il problema iniziale e questo pattern architeturale. Gli spazi di tuple altro non saranno se non gli artefatti con cui gli agenti interagiscono inserendo, rimuovendo e modificando tuple; dietro a questi sarà necessario progettare degli opportuni agenti ambientali che permettano, interagendo con gli spazi di tuple, secondo un certo algoritmo il realizzarsi del comportamento voluto ovvero dell'ordinamento delle tuple.

4.3 Progettazione degli agenti ambientali

4.3.1 Strategie presenti in natura

Un buon modo per approcciare la progettazione degli agenti ambientali è nuovamente quello di osservare in natura da quali comportamenti scaturiscono pattern globali simili a quelli ricercati in questo problema. Come già citato in precedenza il problema del collective sorting è riscontrabile nel *brood sorting* delle formiche e pertanto descrivere come tali insetti si comportano può essere un buon punto di partenza per determinare poi un valido sviluppo per gli agenti. Le formiche si muovono nell'ambiente alla ricerca del cibo e il loro comportamento è essenzialmente caratterizzato da due probabilità, rispettivamente la probabilità di raccogliere un oggetto e quella di lasciarlo. In particolare il comportamento segue un ciclo composto di tre step come segue:

1. raccoglie un oggetto se la concentrazione di tale tipo è minore di quella percepita la volta precedente;
2. inizia a muoversi in maniera casuale;
3. lascia l'oggetto in un luogo che presenta una concentrazione di tale tipo maggiore di quella in l'oggetto è stato raccolto.

In questo modo la completa organizzazione viene raggiunta, spesso in maniera sub-ottima dal punto di vista delle performance visto a livello globale, ma la soluzione resta intrinsecamente robusta.

Chiaramente la soluzione appena discussa deve essere adattata allo scenario del problema rispetto al quale vi sono differenze chiave. Innanzitutto lo scenario studiato non è composto da un ambiente continuo ma da un set di N spazi di tuple fisicamente separati e in secondo luogo a differenza delle formiche gli agenti ambientali non possono muoversi e trasportare tuple ma bensì devono risiedere in ognuno degli spazi citati e provvedere a inviare le tuple agli altri se necessario. In ultimo, la percezione delle formiche che è di carattere solo locale viene contrapposta a quella degli agenti che essendo situati deve

intendendosi con qualsiasi che ogni tupla possiede la stessa probabilità di essere letta.

L'algoritmo degli agenti può essere descritto come segue:

- uno spazio di tuple remoto R viene scelto a caso;
- una urd viene lanciata sullo spazio locale L e restituisce una tupla di tipo T_L ;
- una urd viene lanciata sullo spazio R e restituisce una tupla di tipo T_R ;
- se $T_L \neq T_R$ una tupla di tipo T_R viene spostata (se esiste) dallo spazio L a R .

Maggiore è il numero di un certo tipo K , maggiore sarà la probabilità che la urd ritorni una tupla di quel tipo, ciò fa sì che tramite le due operazioni l'agente si faccia un'idea ipotetica di due buoni aggregatori per le tuple T_L e T_R e per tale motivo l'ultima azione eseguita è nel caso i due tipi siano diversi di mandare una tupla T_R nello spazio R così da migliorare la loro bontà di aggregatori.

4.3.3 Prime simulazioni

Essendo il comportamento degli agenti affetto dalla probabilità, la validazione dell'algoritmo proposto è stata successivamente realizzata tramite una simulazione con il tool MAUDE per controllare che qualsiasi situazione iniziale possa portare a un completo ordinamento. Una prima simulazione è stata realizzata supponendo di avere $N = 4$ spazi di tuple (T1, T2, T3 e T4) e 4 tipi di tuple (K1, K2, K3 e K4) e partendo da una configurazione in cui ogni spazio conteneva 25 tuple per ogni tipo (per un totale di 100 tuple per tipo sui 4 spazi). Alcuni grafici di tale simulazione sono riportati in figura 4.4.

Il grafico 4.4a riporta la tupla vincente per ogni spazio di tuple mostrando come dopo circa 3000 intervalli di tempo viene raggiunta il completo ordinamento in ogni spazio (rappresentato dal raggiungimento della quota 100) anche se l'aggregazione viene raggiunta in tempi diversi nei vari spazi. Il grafico 4.4b, invece, specifica il numero di tuple di ogni tipo nel solo spazio T1, mostrando come la sua dinamica lo porta ad aggregare la tipologia K1 anche se ciò non era assolutamente determinabile a priori tant'è che si può notare come fino all'istante 1000 lo spazio fungeva da buon aggregatore sia per K1 che per K2. Nonostante una simulazione come quella vista dimostri una bontà di fondo dell'algoritmo esistono stati stabili caratterizzati da solo un parziale ordinamento che possono attrarre l'andamento della simulazione e che una volta raggiunti,

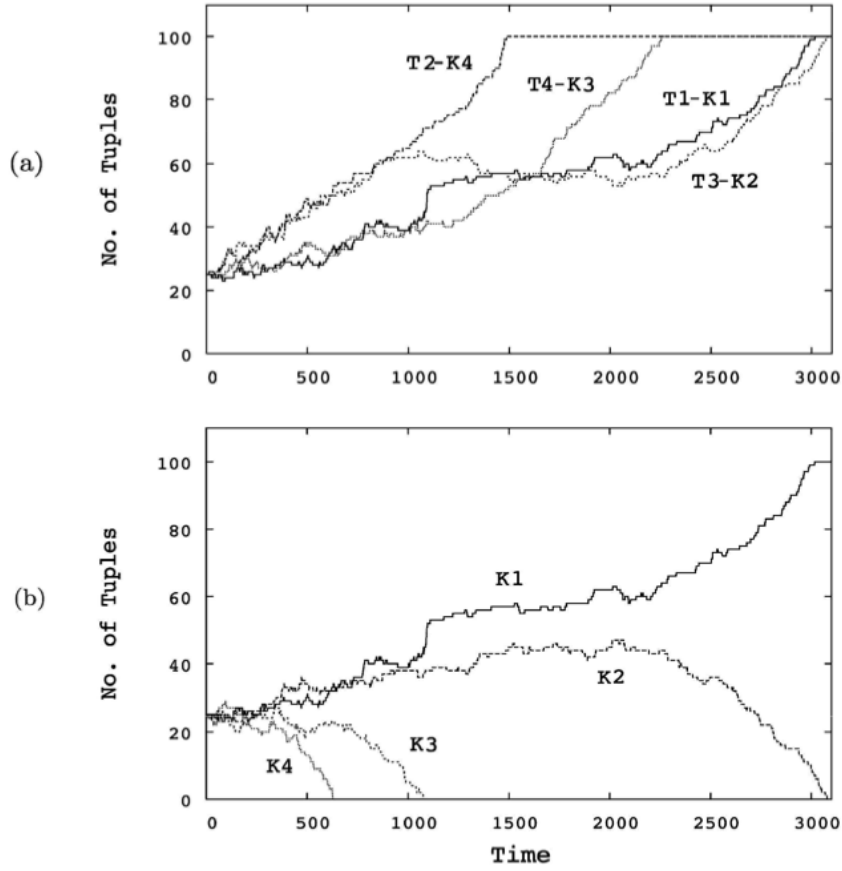


Figura 4.4: Simualzione: (a) tipo vincente in ogni spazio, (b) specifica dello spazio T1.

per come è definito il comportamento degli agenti, è impossibile lasciare. Un esempi di questi minimi locali che possono essere raggiunti è del tipo:

$$T1[100, 0, 0, 0], T2[0, 31, 0, 0], T3[0, 69, 0, 0], T4[0, 0, 100, 100]$$

Nella figura 4.5 si notano le dinamiche che hanno portato (a partire dallo stesso stato iniziale) all'aggregazione delle tuple K2 sia nello spazio T2 che in T3 mentre le tuple di tipo K4 si sono entrambe aggregate nello spazio T4.

4.3.4 Tuning del modello

Il problema è insito nell'impossibilità di poter sfuggire a un minimo locale una volta raggiunto. Il problema può essere nuovamente risolto guardando al mondo animale, infatti nel brood sorting le formiche hanno la concezione di

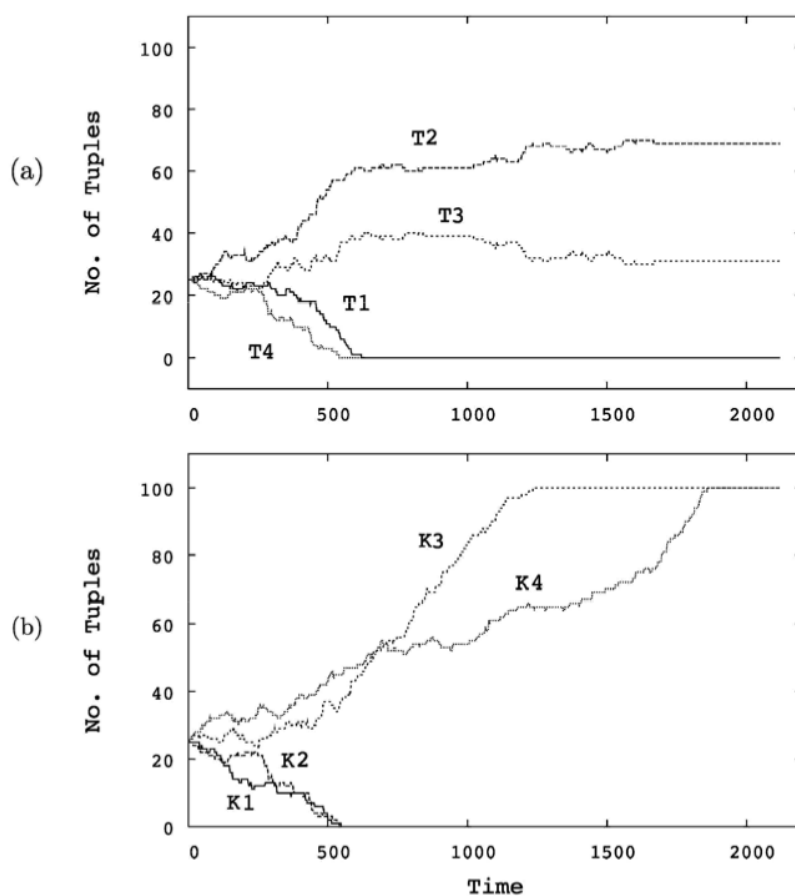


Figura 4.5: Simulazione con ordinamento parziale: (a) divisione di K2 in T2 e T3, (b) specifica dello spazio T4 che aggrega K3 e K4.

concentrazione di un dato oggetto in un luogo ed è vista come rapporto tra volume degli oggetti e unità di spazio, concezione che invece gli agenti non hanno. Per implementare un meccanismo basato su questa idea occorre prevedere l'aggiunta di un ulteriore tipo di tupla detta **noise**. Ora le operazioni di **urd** possono portare a una tupla di tipo **noise** permettendo di determinare un comportamento diverso dell'agente così da sfuggire tramite delle perturbazioni dai minimi locali (una sorta di *simulated annealing*). Non solo, occorre però prevedere la dinamica secondo la quale le tuple **noise** vengono generate e rimosse in modo tale che l'alta presenza di queste sia funzione della vicinanza a un minimo locale, in particolare se due spazi sembra stiano aggregando lo stesso tipo di tuple, il **noise** viene aumentato mentre quando le tuple sono correttamente trasferite il **noise** si decrementa. L'algoritmo riveduto che tiene

conto delle nuove tuple diventa il seguente:

1. uno spazio di tuple remoto R viene scelto a caso;
2. una *urd* viene lanciata sullo spazio locale L e restituisce una tupla di tipo T_L ;
3. una *urd* viene lanciata sullo spazio R e restituisce una tupla di tipo T_R ;
4. se $T_L \neq T_R \neq \text{noise}$ una tupla di tipo T_R viene spostata (se esiste) dallo spazio L a R ;
5. se $T_L \neq T_R = \text{noise}$ una tupla di tipo T_L viene spostata (se esiste) dallo spazio L a R ;
6. se $\text{noise} \neq T_R = T_L$ viene generato una tupla noise in L ;
7. se $\text{noise} \neq T_R \neq T_L \neq \text{noise}$ una tupla noise viene rimossa da L .

Il quarto e quinto punto dell'algoritmo fanno sì che diverse osservazioni in L e R generino sempre un trasferimento. Il sesto punto fa sì che se entrambi gli spazi stanno aggregando le stesse tuple (non di tipo noise) venga aumentato il noise mentre infine il settimo punto fa decrementare il noise se avviene un trasferimento non perturbato.

Simulazioni del nuovo algoritmo, anche a partire da un minimo locale, portano a completo ordinamento nonostante si possa notare un calo delle performance per via dell'introduzione delle nuove tuple noise. Un esempio di tali simulazione è riportato in figura 4.6 a partire dal minimo locale caratterizzato dalla seguente disposizione delle tuple:

$$T1[0, 100, 100, 0], T2[50, 0, 0, 0], T3[50, 0, 0, 0], T4[0, 0, 0, 100]$$

Il grafico 4.6a mostra come nello spazio T2 aumentino le tuple noise determinando a un certo punto un passaggio di tuple K1 da T3 e portando all'aggregazione completa e conseguente diminuzione fino a zero del noise. Il grafico 4.6b, invece, riporta la situazione dello spazio T1 che inizialmente aggrega i tipi K2 e K3 ma in seguito allo svuotamento di T3 le tuple K3 iniziano a defluire portando al completo ordinamento.

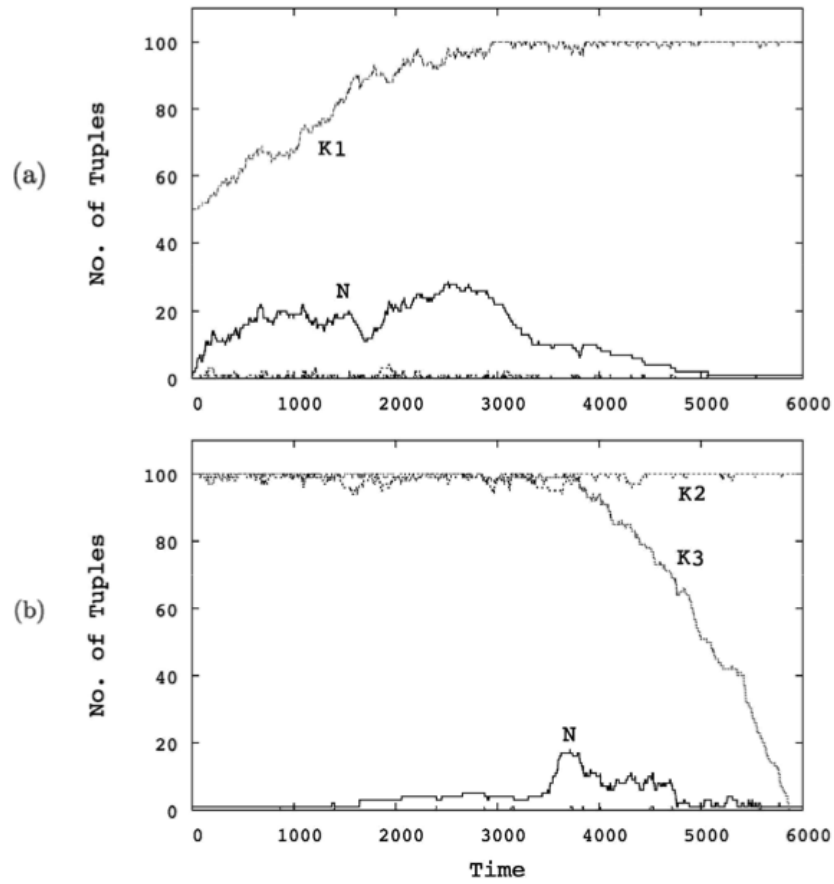


Figura 4.6: Simualzione a partire da un minimo locale: (a) situazione nello spazio T2 con noise in evidenza, (b) specifica dello spazio T1 in cui si nota K3 lasciare lo spazio.

Capitolo 5

Conclusioni

La crescente complessità delle applicazioni necessita di soluzioni composte da sistemi dotati di autonomia, robustezza e adattatività in particolare in nuovi ambiti, quali l'*autonomic computing* o il *pervasive computing*. I sistemi naturali caratterizzati da auto-organizzazione e emergenza posseggono queste qualità e possono essere di ispirazione nell'ideare nuovi meccanismi per risolvere tali problemi.

Essendo l'auto-organizzazione e l'emergenza fenomeni che si rilevano a un macro livello, ma generati dalle interazioni che avvengono a livelli più bassi tra entità semplici, come per esempio avviene per gli insetti sociali, un buon modo per riprodurre tali strutture è l'utilizzo dei **sistemi multi-agente**. I possibili meccanismi utilizzabili sono diversi e si suddividono principalmente in quattro classi: basati su interazioni e comunicazioni dirette tra gli agenti; basati su stigmergia e interazioni indirette; meccanismi basati sul rinforzo dove gli agenti cambiano il proprio ruolo a runtime; e agenti dal comportamento cooperativo.

Quale che sia il meccanismo utilizzato, la difficoltà maggiore risiede nella modellazione del comportamento degli agenti che nel complesso deve far emergere il comportamento globale e a cui non ci si può approcciare utilizzando le tradizionali tecniche della software engineering non essendo l'obiettivo globale predicibile come funzione o somma di quelli locali. Per tale motivo l'affidarsi anche a fasi di simulazione, come si è visto nel caso di studio, risulta vitale per convalidare e valutare la convergenza delle soluzioni sviluppate. A ciò occorre aggiungere che, come in natura, anche il ruolo dell'ambiente è spesso importante e deve essere ben compreso e non sottovalutato poiché parte integrante da modellare come nel caso della stigmergia.

Bibliografia

- [1] L. Gardelli, M. Viroli, M. Casadei, A. Omicini, *“Designing self-organising environments with agents and artefacts: a simulation-driven approach”* DEIS, Alma Mater Studiorum, Università di Bologna, 2005
- [2] H. Van Dyke Parunak, S. Brueckner *“Entropy and Self-Organization in Multi-Agent Systems”* Proceedings of the International Conference on Autonomous Agents, 2001
- [3] C. Sansores, J. Pavón *“An Adaptive Agent Model for Self-Organizing MAS”* Universidad Complutense Madrid, Facultad de Informática, 2008
- [4] J. M. E. Gabbai, Hujun Yin, W. A. Wright and N. M. Allinson *“Self-Organization, Emergence and Multi-Agent Systems”* Dept of EE&E, University of Manchester, 2005
- [5] A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, L. Tummolini *“A Conceptual Framework for Self-Organising MAS”* Workshop from Object to Agents, Torino 2004
- [6] M. Morandini, F. Migeon, M. P. Gleizes, C. Maurel, L. Penserini, A. Perini *“A Goal-Oriented Approach for Modelling Self-Organising MAS”* International Workshop “Engineering Societies in the Agents World” (ESAW 09) 2009

Elenco delle figure

1.1	I termitai sono un esempio di auto-organizzazione tramite stigmergia.	3
1.2	Banco di pesci in nuoto coordinato.	3
2.1	L'emergenza del percorso minimo tra tana e sorgente di cibo in seguito all'interazione stigmergica delle formiche.	7
4.1	I componenti del metamodello Agenti & Artefatti.	14
4.2	Pattern architetturale che prevede la presenza di agenti ambientali per introdurre funzionalità auto-organizzanti.	14
4.3	Scenario finale	16
4.4	Simualzione: (a) tipo vincente in ogni spazio, (b) specifica dello spazio T1.	18
4.5	Simualzione con ordinamento parziale: (a) divisione di K2 in T2 e T3, (b) specifica dello spazio T4 che aggrega K3 e K4. . . .	19
4.6	Simualzione a partire da un minimo locale: (a) situazione nello spazio T2 con noise in evidenza, (b) specifica dello spazio T1 in cui si nota K3 lasciare lo spazio.	21