

# UNIVERSITA' DEGLI STUDI DI TORINO

Facoltà di Economia  
Corso di laurea in Economia e Commercio

## TESI DI LAUREA

Analisi di un mercato di borsa:  
Una simulazione con agenti, utilizzando i classifier system.

Relatore: Professor Pietro Terna  
Correlatore: Professor Sergio Margarita

Candidato:  
Claudia Negro

ANNO ACCADEMICO 1999/2000.

# Indice

Introduzione	III
Capitolo Primo: Complessità.	1
1.1.1 – Cibernetica.	2
1.1.2 – Teoria delle Catastrofi.	3
1.1.3 – Caos deterministico.	4
1.1.4 – Frattali.	5
1.1.5 – Complessità.	7
1.2 – Complessità e comportamenti emergenti.	8
1.2.1 – Sistemi non lineari.	9
1.2.2 – Comportamenti emergenti.	10
1.2.3 – Razionalità induttiva.	11
1.2.4 – L'esperimento di El Farol: un esempio da Santa Fe.	12
1.2.5 – Analisi del problema.	13
Bibliografia.	15
Capitolo Secondo: Il sistema economico come sistema complesso.	17
2.1.1 – Mercati reali e mercati artificiali.	20
2.1.2 – Complessità nei mercati.	21
2.1.3 – Il mercato azionario come sistema complesso.	23
2.1.4 – ASM, il modello del SFI: un esempio.	26
Bibliografia.	32
Capitolo Terzo: Intelligenza Artificiale.	33
3.1 – Cognitivismo e Connessionismo.	34
3.1.1 – Il test di Torino.	36
3.1.2 – Due esempi di elaborazioni sulla base del test di Turing: il caso di Eliza il dottore e Parry il paranoico.	36
3.1.3 – Simboli e significati.	37
3.1.4 – Il test della stanza cinese.	38
3.1.5 – Il ragionamento e la soluzione dei problemi.	40
3.1.6 – I sistemi esperti.	41
3.1.7 – Le reti neurali artificiali (RNA).	42
3.2 – Dall'Intelligenza Artificiale all' <i>Alife</i> .	45
3.2.1 – L'evoluzionismo.	46
3.2.2 – Algoritmi genetici.	46
3.2.3 – <i>Classifier Systems</i> .	49
3.2.3.1 – Struttura e funzionamento.	49
3.2.3.2 – I premi e le penalità, l'algoritmo <i>bucket brigade</i> .	50
3.2.4 – Automi cellulari.	51
3.2.5 – Tierra e Life, due esempi.	52
Bibliografia.	55

Capitolo Quarto: Modelli di simulazione basati su agenti.	56
4.1.1 – <i>Swarm</i> e la sperimentazione nelle scienze sociali.	59
4.1.2 – Struttura e funzionamento di un modello in <i>Swarm</i> .	61
4.1.2.1 – Creazione degli oggetti.	62
4.1.2.2 – Lo schema ERA ( <i>Environment, Rules, Agents</i> ).	64
4.1.3 – La programmazione ad oggetti.	66
Bibliografia	68
 Capitolo Quinto: un mercato di Borsa.	69
5.1.1 – Sum, <i>Surprising (Un)realistic Market</i> .	70
5.1.2 – Agenti con e senza “mente”.	70
5.1.3 – Il <i>book</i> ed il meccanismo di formazione del prezzo.	72
5.1.4 – I parametri del modello SUM.	74
5.1.5 – Esperimenti con agenti di tipo <i>random</i> .	77
5.1.6 – Imitatori locali, imitatori del mercato ed agenti <i>stop loss</i> .	81
5.1.7 – Analisi del mercato sotto il profilo della previsione.	86
5.1.8 – Il metodo dei <i>cross target</i> .	89
Bibliografia	98
 Capitolo Sesto: il modello SUM con l'uso dei classifier system	99
6.1.1 – Il pacchetto CW	100
6.1.2 – Applicazione dei classifier system al modello SUM	107
6.1.3 – Esperimenti, risultati e conclusioni	115
Bibliografia	129
 Bibliografia	130

## Introduzione

La realtà è fatta di sistemi complessi, cioè di sistemi nei quali molte cause concorrono a creare un solo effetto e nei quali le relazioni fra le cause stesse sono non lineari, nel senso che l'effetto di ogni singola causa non è indipendente da quello delle altre, per cui non può essere isolato.

I sistemi complessi sono generalmente composti da elementi eterogenei che si influenzano localmente; ogni individuo appartenente al sistema interagisce solo con un ristretto numero di altri individui e, da queste interazioni locali, emergono fenomeni globali complessi, che non sono deducibili a priori, pur conoscendo i singoli elementi ed i legami di interazione che intercorrono fra loro.

Nel corso dell'ultimo secolo si è riscontrata una crescente attenzione per i fenomeni cosiddetti complessi da parte di molte discipline e, questo fattore, ha reso molto difficoltoso il percorso di economisti, biologi, matematici ed informatici verso una definizione del problema, prima ancora che verso l'individuazione del mezzo migliore per affrontarlo.

In precedenza l'attenzione si era focalizzata soprattutto sui sistemi semplici, perché tali sistemi si prestano ad essere studiati con gli strumenti matematici e sperimentazioni tradizionali, come le equazioni ed i sistemi di equazioni lineari, che, però, diventano quasi impossibili da applicare ai sistemi complessi, a causa delle loro numerose variabili e le complicate interazioni che intercorrono fra le variabili stesse.

Inoltre, i sistemi complessi hanno delle componenti dinamiche e di imprevedibilità che gli strumenti analitici tradizionali hanno difficoltà ad affrontare.

Il contrasto tra sistemi complessi e strumenti tradizionali della scienza diventa molto evidente nell'uso e nel metodo degli esperimenti di laboratorio; infatti, gli strumenti che si adattano alle scienze naturali, risultano totalmente inadeguati per le scienze sociali.

Le simulazioni e, in particolare, le simulazioni condotte mediante modelli ad agenti adattivi rappresentano il principale strumento, per mezzo del quale è possibile condurre esperimenti su fenomeni delle scienze sociali.

Il principale vantaggio delle simulazioni condotte mediante esperimenti al computer, consiste nel fatto che lo sperimentatore può inserire gradualmente nel programma le diverse parti della teoria, modificare i dati già presenti e verificare il funzionamento della teoria, osservando i risultati della simulazione.

Le simulazioni di mondi artificiali consentono indagini più semplici ed approfondite rispetto alla semplice analisi della realtà, poiché tempo e causalità sono determinati dallo sperimentatore, le caratteristiche ambientali possono essere modificate, la rilevazione dei dati è più precisa e l'esperimento può essere ripetuto a piacere.

La possibilità di modificare la velocità di scorrimento del tempo contribuisce a ridurre la durata della ricerca che, nella realtà, può richiedere la ripetizione periodica delle osservazioni, impegnando tempi considerevolmente lunghi; il controllo sulla manifestazione degli eventi casuali consente di

produrre dati immediatamente utilizzabili, senza bisogno di procedere ad elaborazioni preventive necessarie per eliminare le componenti di disturbo quando si utilizzassero dati reali.

Nelle scienze sociali, l'osservazione dei fenomeni viene spesso effettuata tramite interviste o valutazione dei dati forniti dagli istituti di statistica: in questi casi occorre determinare preventivamente quali dati rilevare e, successivamente, considerare margini di errore derivanti da cattiva comprensione dei formulari o da risposte insincere.

Impiegando agenti virtuali, invece, tutti i dati possono essere rilevati facilmente, si evitano errori e non occorre determinare a priori quali grandezze osservare, inoltre, casi particolari, o estremi, si manifestano difficilmente in ambiti reali, e dato che la ripetizione della rilevazione è onerosa e le variazioni delle condizioni al contorno non sempre rispondono alle esigenze del ricercatore, possono nascere problemi che non si hanno con le simulazioni.

L'indagine su ambiti virtuali risulta, perciò, meno costosa, più agevole, maggiormente adattabile agli scopi della ricerca e capace di fornire risultati fortemente innovativi, infatti, con i modelli basati su agenti si possono ottenere risultati interessanti, grazie al loro elevato grado di indipendenza e alla capacità di adattare il loro comportamento alle caratteristiche dell'ambiente in cui operano.

Fenomeni complessi, di una certa importanza, risultano emergere già con l'impegno di oggetti capaci di reagire, in modo prestabilito, a stimoli semplici; il comportamento di simili agenti risulta, però, ancora influenzato dalle scelte di chi redige il programma.

Quando la determinazione delle regole di comportamento avviene in base a tecniche di intelligenza artificiale, invece, l'azione di ogni agente risulta essere conseguenza del grado di apprendimento del medesimo, permettendo di garantire l'autonomia sufficiente a promuovere lo sviluppo di fenomeni innovativi.

Le tecniche di intelligenza artificiale, più conosciute ed adatte allo scopo, fanno capo ai due paradigmi delle reti neurali artificiali e dei metodi evolutivi: algoritmi genetici, *classifier system* e *genetic programming*.

Il presente lavoro tratta dell'impiego dei metodi evolutivi, con particolare riferimento ai *classifier system*, una tecnica che si basa sull'imitazione del processo di evoluzione neurale: le regole sono assoggettate a procedimenti di riproduzione ed estinzione, atti a garantire la sopravvivenza di quelle che permettono di ottenere i migliori risultati. Si procede valutando ciascuna regola per stabilire quali, fra esse, dovranno contribuire alla formazione di nuove istanze, cioè riprodursi, e quali saranno sostituite dalle nuove regole. La ripetizione ciclica del procedimento consente un graduale perfezionamento del patrimonio normativo e le mutazioni delle condizioni ambientali, comportano variazioni del "valore" di ciascuna regola, vengono continuamente recepite dal sistema con evoluzioni delle regole stesse.

Nella seconda parte del lavoro si tratterà dell'applicazione di algoritmi evolutivi, in particolare di *classifier system*, ad un mercato di borsa, all'interno del quale operano agenti eterogenei, dotati di diversi gradi di razionalità.

Nella versione base del modello, infatti, operano solo agenti che agiscono casualmente, coordinati dalla struttura di un *book* elettronico, strutturato sul modello di quello del mercato telematico della Borsa di Milano.

Successivamente sono stati introdotti agenti *chartisti*, di tipo *stop loss*, imitatori del mercato ed imitatori locali, per valutarne gli effetti all'interno del mercato, sia per quanto riguarda la stabilità, sia per quanto riguarda la prevedibilità.

Infine, sono stati introdotti degli agenti dotati di aspettative razionali e costruiti con il metodo dei Cross Target.

Questa metodologia serve per sviluppare un meccanismo di apprendimento, che si basa sull'applicazione delle reti neurali artificiali, sulla base del quale gli agenti imperano ad essere coerenti con i *target* sui quali viene esercitata la rete neurale, correggendo contemporaneamente le congetture effettuate sulle azioni da compiere e sugli effetti di tali azioni.

L'ultimo passo è consistito nell'introduzione di agenti che utilizzano i *classifier system* per autosviluppare le proprie regole di comportamento all'interno del mercato di borsa.

Quest'ultimo tipo di agenti è stato utilizzato al fine di valutarne gli effetti sulla stabilità del mercato e sul livello di prevedibilità, ma anche per valutare l'efficacia dell'algoritmo evolutivo nel generare regole sufficientemente valide da permettere agli agenti di arricchirsi e di reagire alle eccessive variazioni dei prezzi, che si verificano in corrispondenza delle bolle speculative e dei *crash* del mercato borsistico.

I risultati ottenuti possono essere considerati decisamente buoni, infatti, gli agenti dotati di autoapprendimento si sono dimostrati in grado di sviluppare regole di comportamento valide a reagire correttamente agli input del mercato e ad incrementare il proprio livello di ricchezza, mostrando una notevole varietà di strategie e, soprattutto, di saper gestire le proprie dotazioni di liquidità e di azioni per tentare di trarre un guadagno.

## Capitolo Primo: Complessità.

“Ciò che è  
senza  
ambiguità e senza contraddizioni  
coglie soltanto un lato delle cose.”

C. G. Jung, *Psicologia  
ed Alchimia*.

Nel corso dell'ultimo secolo si è riscontrata una crescente attenzione per i fenomeni cosiddetti complessi, da parte di molte discipline e questo fattore, sicuramente positivo, ha tuttavia reso molto mal definito il campo di studio e, conseguentemente, ha reso difficoltoso il percorso di economisti, biologi, matematici ed informatici verso una definizione del problema, prima ancora che verso l'individuazione del mezzo migliore per affrontarlo.

Walrop (1995) riferendosi al gruppo di studio del Santa Fe Institute, dedicato in modo particolare allo studio sulla complessità, dichiara: “ *they all share the vision of an underlying unity, a common theoretical framework for complexity that would illuminate nature and humankind alike [...] They believe that their application of these ideas is allowing them to understand the spontaneous, self-organizing dynamics of the world in a way that no one ever has before – with the potential for immense impact on the conduct of economics, business, and even politics...They believe they are creating, in the words of Santa Fe institute, «the sciences of the twenty-first century».*

Dal versante opposto, Horgan (1995,1997), sostiene che la complessità rappresenta soltanto l'ultimo di una serie di discipline che hanno rappresentato scarsi tentativi di spiegazione dei fenomeni emergenti, la complessità, secondo Horgan non riesce a superare lo scoglio della non linearità dei fenomeni, come i suoi predecessori, cibernetica, teoria delle catastrofi e caos.

In Horgan (1997) *"So far chaoplexologists have created some potents metaphors: the butterfly effect, fractals, artificial life, the edge of chaos, self organized criticality. But they have not told us anything about the world that is both concrete and truly surprising, either in negative or in a positive sense."*

Le due posizioni estreme presentate, forniscono un'idea abbastanza precisa di come il mondo scientifico sia in contrasto, non solo per quanto riguarda il metodo o i risultati di studio, ma addirittura sull'opportunità di tali studi e su come la problematica risulti mal definita.

In Day (1994) si trova un tentativo di definizione di problema complesso: *"A dynamical system is complex if it endogenously does not tend asymptotically to a fixed point, a limit cycle, or an explosion. Such system can exhibit discontinuous behavior and can be described by sets of nonlinear differential or difference equations, possibly with stochastic elements. But not all such equation systems will generate complexity. The positive exponential function, basis for most growth models, is an example of a noncomplex, nonlinear system because it just explodes"*

Il tentativo di definizione offerto da Day è particolarmente interessante perché comprende oltre a ciò che attualmente è definito complessità, anche i suoi predecessori, la cibernetica, la teoria delle catastrofi, il caos e la geometria frattale, dando in parte ragione alle critiche di Horgan.

Benché la complessità sia un concetto multidisciplinare, e derivi essenzialmente dalla matematica e dalla fisica, le complicità che si riscontrano nell'applicazione del concetto all'economia, a causa dell'interazione degli individui all'atto di prendere decisioni, aggiungono alla complessità, non così come intesa da Day, un grado ulteriore, che non esiste nelle altre discipline che l'hanno preceduta.

### **1.1.1 - Cibernetica.**

La Cibernetica è una disciplina scientifica fondata da Norbert Wiener nella seconda metà degli anni Quaranta, che coniò questo termine pensando ad un sostantivo che fosse appropriato per descrivere il campo del controllo: gli venne allora in mente la parola "timoniere", che in greco si dice κυβερνήτης, e passando all'idioma anglosassone, risultò il neologismo *cybernetics*.

La cibernetica, infatti, è la scienza che si occupa del controllo e della regolazione dei sistemi complessi, senza distinguere fra macchinari o sistemi viventi. In particolare Wiener si occupò del concetto di *feedback*, cioè del processo che consiste nel trasferire informazioni dall'uscita all'ingresso di un sistema allo scopo di ottenerne un particolare comportamento.

La cibernetica, dunque, studia i meccanismi di comunicazione e di interazione fra gli esseri viventi, sia allo scopo di aumentare le conoscenze di tali meccanismi, sia per derivarne modelli che possono essere riprodotti artificialmente.



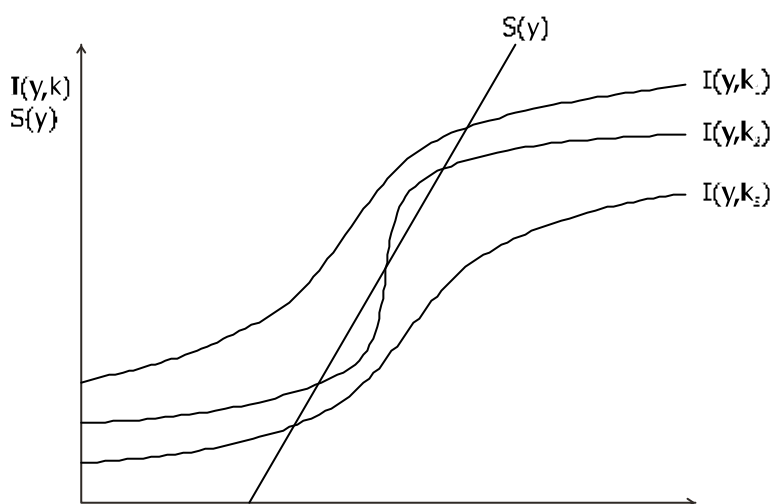
In ambito economico, Forrester (1961), applica le teorie della cibernetica ai sistemi industriali e pone le basi per uno studio dinamico dei sistemi, dando risalto ai risultati sorprendentemente controintuitivi e discontinui che possono sorgere da sistemi di equazioni non lineari.

Tali idee hanno trovato un senso nei modelli attuali di complessità, compresa l'auto-organizzazione e l'emergenza di strutture dal basso verso l'alto, come l'emergenza dei modelli di vicinato nei distretti residenziali di una città, dalle decisioni di localizzazione dei singoli agenti.

### 1.1.2 - Teoria delle Catastrofi.

Il matematico francese René Thom (1975) sviluppò la teoria delle catastrofi, individuando nelle catastrofi dei particolari punti di discontinuità dei sistemi dinamici.

Le discontinuità dipendono da equilibri multipli distinti e procedono saltando da uno ad un altro a seconda di come variano gradualmente alcuni parametri.



**Fig.1.1** – Funzioni di Kaldor non lineari degli investimenti con equilibri multipli.

La figura mostra una versione dovuta a Varian (1979) del ciclo di Kaldor (1940), in cui il risparmio ( $S$ ) è una funzione lineare del reddito ( $y$ ). L'investimento ( $I$ ) è una funzione dalla forma ad "S" del reddito e del livello di capitale esistente ( $K$ ). In equilibrio, vale la seguente equazione:  $S = I$ .

Chiaramente la funzione di investimento intermedia permette l'esistenza di più di un equilibrio e si potrebbe passare da un equilibrio all'altro a seconda che la funzione salga o scenda lungo il ciclo. Simili comportamenti e dinamiche si riscontrano nei modelli di dinamiche discontinue nell'equilibrio fra occupazione urbana e rurale (Casetti, 1980) e nei mercati di scambio esteri (Krugman, 1984).

La teoria delle catastrofi generò un entusiasmo anche maggiore della cibernetica in molte discipline. Infatti, i modelli di discontinuità rimangono uno dei temi maggiormente affrontati nei più recenti modelli complessi.

Nonostante questo aspetto, la teoria delle catastrofi fu duramente criticata, poiché venne utilizzata per spiegare ogni genere di discontinuità, anche quando, per poterla applicare, occorrevano eccessive restrizioni alle ipotesi.

Le maggiori critiche all'applicazione della teoria delle catastrofi in economia vennero da Zahler e Sussman (1977), i quali focalizzarono la propria attenzione sul modello dinamico di mercato azionario di Zeeman (1974). In questo modello, si osservavano dei *crash* quando i *traders chartisti* erano in numero maggiore rispetto a quelli che agivano secondo l'analisi fondamentale.

In quel momento la critica di Zahler e Sussman, che individuava eventi catastrofici in presenza di agenti irrazionali e non scientifici nei loro comportamenti, a molti economisti sembrò abbastanza convincente da far sì che la teoria delle catastrofi venisse abbandonata.

### **1.1.3 - Caos deterministico.**

La teoria del Caos nasce negli anni '60 grazie agli studi del meteorologo E.N. Lorenz, il quale propose un semplice sistema di tre equazioni, per costituire un modello che rendesse possibile studiare i moti convettivi degli strati d'aria. Queste equazioni, perfettamente deterministiche, avevano soluzioni caotiche.

Nel 1975 la comunità scientifica, nella figura del professor James Yorke, utilizzò, per la prima volta, il termine caos, riferendosi all'insieme dei fenomeni generati da quelli che oggi si individuano come sistemi dinamici caotici.

Inizialmente il caos era stato confuso con la casualità e l'indeterminatezza; molti sistemi, infatti, apparivano casuali e quindi si pensava che fossero effettivamente tali, questo nonostante il fatto che alcuni di essi mostrassero fasi in cui il comportamento era quasi periodico e fasi in cui il comportamento era casuale. Questa osservazione portò ad una possibile definizione di caos come sovrapposizione di un numero elevato di moti periodici instabili.

Ciò che invece caratterizza un sistema dinamico caotico, risiede nei concetti di *transitività topologica* (le orbite visitano l'intero attrattore) e di *dipendenza sensibile dalle condizioni iniziali* (due orbite, con condizioni iniziali vicinissime, tracciano, in un primo tempo, traiettorie pressoché sovrapposte per poi seguire, da un dato istante in poi, ciascuna un cammino differente).

Considerato un sistema qualunque, ad esempio un pendolo, si nota che, trascorso un tempo  $T$ , il sistema pendolo raggiunge una situazione di equilibrio di tipo oscillatorio caratterizzata da un'orbita ripetitiva con un periodo ben definito; questa situazione di equilibrio viene denominata attrattore del sistema.

L'oscillatore scoperto da Lorenz era caratterizzato da oscillazioni aperiodiche caotiche, anche se il sistema era caratterizzato da equazioni strettamente deterministiche, il meccanismo dal quale originava il caos era determinato da perturbazioni microscopiche amplificate fino ad interferire con il comportamento macroscopico del sistema stesso.

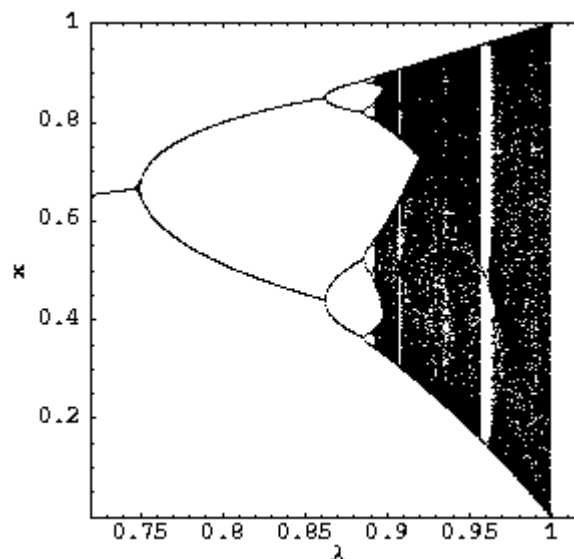
Spesso per illustrare un comportamento caotico ed introdurre i concetti di base del caos viene utilizzata l'equazione logistica:

$$X_t = kX_{t-1}(1 - X_{t-1}), \text{ in cui } k \text{ è}$$

una costante

Questa equazione può essere usata per illustrare il modello di diffusione del contagio di una malattia, infatti se il tasso di propagazione del contagio è determinato da  $k$ , allora la diffusione del contagio è proporzionale alla percentuale di popolazione che è già stata contagiata, determinata da  $X$ . Ma la diffusione del contagio risulta anche essere proporzionale ad  $(1 - X)$ , cioè al numero di coloro che non sono ancora stati contagiati, come in un modello predatore-preda, i due gruppi sono legati vicendevolmente da un ciclo iterativo complesso.

Secondo il concetto di *feedback*, nell'equazione logistica, il risultato delle iterazioni precedenti viene reimmesso nel ciclo successivo.



**Fig. 1.2** - Biforcazione logistica.

#### 1.1.4 - Frattali.

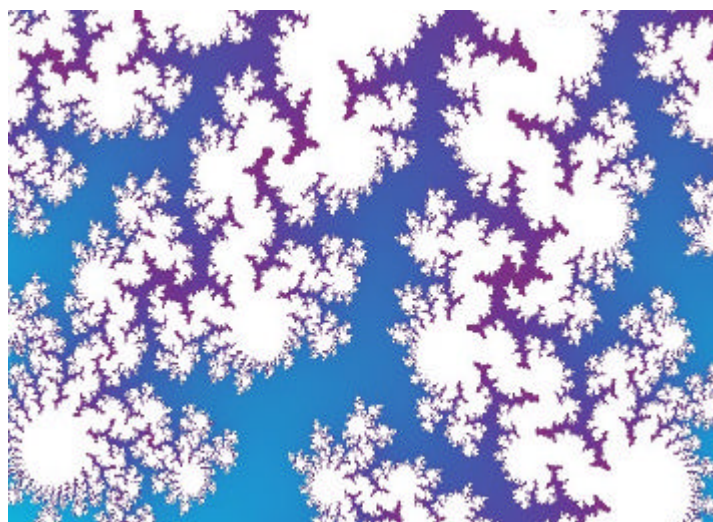
In Edgar e Peters (1991) *"With fractals, mathematicians have created a system that describes natural shapes in term of a few simple rules. Complexity emerges from this simplicity. Fractals give structure to complexity and beauty to chaos."*

In natura, non solo i fenomeni oscillanti, ma anche le forme mostrano sempre una caratteristica commistione di ordine e di irregolarità, che assume frequentemente l'aspetto di strutture simmetriche (ma mai totalmente speculari), di fini suddivisioni e ramificazioni (ma mai totalmente geometriche), di somiglianze tra organismi distinti (ma mai completamente sovrapponibili). Anche questo tipo di "ordine nascosto" fa parte dei fenomeni caotici e va sotto il nome di geometria frattale.

La geometria frattale è la geometria che meglio si adatta allo studio del caos. Il termine fu coniato nel 1975 da B.B. Mandelbrot ed acquisì ampia notorietà nel mondo scientifico a partire dai primi anni '80 Mandelbrot (1982). Le forme frattali possono essere generate al calcolatore con algoritmi più o meno complessi. Mediante queste operazioni, appaiono figure bi o tridimensionali con gran varietà di forme diverse, presenza di sottili ramificazioni che si possono seguire nei più fini dettagli ed autosomiglianza, in modo che ingrandendo una parte della struttura è possibile rilevare dettagli che si ripetono a diverse scale di ingrandimento. Molte figure di questo tipo, specialmente se eseguite a colori, appaiono anche particolarmente belle, presentando un insieme attraente di varietà e di regolarità.

Anche l'attrattore di Lorenz ha una tipica struttura a frattale: le linee delle traiettorie che esso disegna formano delle bande finemente ed irregolarmente spaziate, secondo uno schema a frattale che si osserva su qualunque scala di ingrandimento e di definizione. Per osservare un maggior numero di bande, è sufficiente aumentare il numero delle iterazioni. Molti oggetti naturali, apparentemente disordinati, godono di queste proprietà, per cui i frattali sono strumenti utili per descrivere una gran varietà di fenomeni fisici e di forme naturali: nel mondo inanimato sono frattali i fulmini, il delta dei fiumi, molti tipi di aggregati cristallini inclusi i fiocchi di neve, le configurazioni di fluidi non miscibili, gli strati sedimentari che costituiscono montagne quali le Dolomiti; in biologia, gli esempi di frattali si moltiplicano e comprendono le strutture ramificate dei vasi sanguigni ed il sistema di Purkinje che conduce i segnali elettrici nel cuore.

Nella formazione degli oggetti con dimensioni frattali si assiste ad una particolare interazione tra eventi stocastici (casuali) ed eventi determinati dallo stato del sistema fisico che va crescendo. Tale tipo di crescita, detto anche *aggregazione per diffusione*, molto probabilmente ha avuto un ruolo fondamentale nella nascita della vita sulla terra e lo ha continuamente nei processi di accrescimento fisici e biologici.



**Fig.1.3** – Frattale

### 1.1.5 - Complessità.

Il termine "complesso" indica un oggetto o uno stato costituito da elementi affini ed eterogenei associati in maniera non districabile, al contrario, un oggetto "complicato" comprende parti semplici e divisibili, connesse da rapporti lineari evidenti o, se impliciti, evidenziabili. La complessità, dunque, non può essere concepita senza considerare le parti che la compongono, né le parti possono essere comprese senza considerare l'intero.

Un sistema complesso è un sistema in cui sono diversamente coinvolti numerosi fattori indipendenti, legati da numerose interazioni, la cui varietà permette al sistema stesso nella sua totalità, di sviluppare un'auto-organizzazione spontanea; dall'azione contemporanea di singoli individui che perseguono fini particolari, cercando di soddisfare i propri bisogni, nasce un'organizzazione complessa, senza che nessuno degli individui coinvolti nel processo ne sia la guida o sia consapevole del risultato delle sue azioni semplici.

Le molecole che compongono un corpo vivente, sicuramente non hanno percezione del corpo stesso, né della vita che lo anima, l'iter evolutivo che ha portato gli amminoacidi fondamentali all'aggregazione in cellule viventi corrisponde all'evoluzione di un sistema complesso adattivo auto-organizzato, tuttavia le trasformazioni che hanno portato dalla "non-vita" alla vita sono tutt'altro che spiegate, proprio perché non è rappresentabile con esattezza l'insieme delle reazioni che legano tutte le componenti del sistema.

Langton (1989), nel corso dei suoi studi sugli automi cellulari, che, data la loro struttura, si prestano particolarmente a rappresentare modelli legati alla chimica ed alla fisica elementare, individuò nella transizione di fase, che venne successivamente definita "margine del caos", l'elemento che rende possibili la vita e la mente, cioè un equilibrio tra le forze dell'ordine e quelle del disordine.

In quest'ottica diviene fondamentale focalizzare l'attenzione sul comportamento dei sistemi, piuttosto che sulle loro componenti, in questo modo si individuano i due estremi dell'ordine e del caos. Fra i due estremi, nella fase del margine del caos, si trova anche la complessità, una classe di comportamenti in cui le componenti del sistema non raggiungono una posizione stabile, ma neanche si dissolvono nella turbolenza; questi sistemi sono abbastanza stabili da memorizzare informazione, ma anche sufficientemente labili da trasmetterla, sono caratterizzati da un dinamismo che li rende spontanei, adattivi, disordinati, ma non riconducibili alle dinamiche caotiche.

In Waldrop (1995) "Il margine del caos è là dove nuove idee e genotipi innovativi erodono senza tregua i confini dello *status quo* (...), è il campo di battaglia perennemente in bilico tra inerzia e anarchia, l'unico luogo in cui un sistema complesso può essere spontaneo, adattivo e vivo."

Morin (1997) sostiene che esistono tre principi per pensare la complessità, un principio dialogico, che permette di sintetizzare due termini al contempo antagonisti e complementari, distinguendo e non disgiungendo, considerando gli opposti come estremi di un *continuum* e non opposizioni logiche non rapportabili.

Il secondo è il principio della ricorsività dell'organizzazione, secondo il quale gli effetti sono al contempo causa e produttori di ciò che li genera. Il terzo principio è quello ologrammatico, la complessità, come un ologramma, contiene in ogni suo punto la quasi totalità dell'informazione su se stessa.

Quest'interpretazione, in particolare, prende le distanze dalle due posizioni estreme rispetto alla complessità, l'una che privilegia le parti e si può ricondurre al pensiero riduzionista e l'altra che trascura, invece, le parti in favore del tutto e fa riferimento alla corrente olista.

Da queste definizioni emerge come l'incertezza costituisca un elemento fondamentale della complessità e del suo studio, qualunque aspetto si voglia privilegiare, quello maggiormente psicologico di Morin (1997) e Bateson (1976) o quello sociale ed economico del Santa Fe Institute.

## **1.2 - Complessità e comportamenti emergenti.**

I sistemi complessi adattivi, così come li ha concepiti Holland, sono gruppi di agenti legati in un processo di co-adattamento, in cui le mosse di adattamento di ciascuno hanno conseguenze per l'intero gruppo di individui. Holland (1993) e Kauffman (1993), mostrano che, sotto determinate condizioni, modelli semplici di questo processo presentano sorprendenti capacità di auto-organizzazione.

L'emergenza della complessità organica nella società umana è oggetto di studio per discipline molto diverse, e la simulazione al computer è il mezzo che si è dimostrato il migliore per unificarne il metodo di studio, infatti questi modelli consistono in mondi artificiali, abitati da popolazioni di individui eterogenei, i quali interagiscono l'uno con l'altro e modificano il proprio ambiente.

Un esempio di questo tipo di modelli è "Sugarscape", sviluppato da Axtell ed Epstein (1996) *"Sugarscape is a simulated landscape consisting of a lattice with resource-bearing sites. Population of "adaptive agents" are created and endowed with internal states, behavioral rules, and the capability to move around on the landscape. Some internal states are fixed for the agent's life span, while others change in response to interaction with other agents or with the external environment. Sugarscape has been used to test hypotheses about economic, psychological, and social behavior, and spread of disease."*

Su un livello più astratto, la teoria della complessità offre nuovi strumenti per esplorare alcuni dubbi e presupposti per l'emergenza della cooperazione.

Per esempio, Lindgren e Nordhal (1994) hanno studiato l'evoluzione della cooperazione in giochi durante i quali i giocatori erano isolati e potevano comunicare solo con i loro immediati vicini. In una versione semplice di quest'esperimento, gli individui giocavano un gioco a due come il Dilemma del prigioniero reiterato, decidendo di cooperare ciascuno con il proprio vicino. Il risultato interessante fu che emersero aree spaziali di cooperazione e di defezione, che aumentavano i *payoff* per tutti i giocatori. In questo modo, un sistema complesso adattivo può emergere dall'interazione di collezioni eterogenee di agenti.

### 1.2.1 - Sistemi non lineari.

La definizione di sistema non lineare, ottenuta per negazione della definizione di sistema lineare, dice che “un sistema non lineare è un sistema il cui comportamento non è uguale alla somma delle singole parti che lo compongono”.

Se, dunque, per procedere allo studio dei sistemi lineari, si procede alla loro scomposizione ed allo studio analitico di ciascuna delle parti, questo non può avvenire per lo studio dei sistemi non lineari.

Il comportamento di tali sistemi, infatti, dipende dall'interazione delle parti, più che dal comportamento delle parti stesse, occorre quindi considerare il sistema non lineare come un tutto non uguale alla somma delle parti e, dunque, occorre focalizzarsi sulle dinamiche di interazione fra gli elementi che compongono il sistema.

Data questa caratteristica dei sistemi non lineari appare abbastanza evidente il loro legame con la complessità ed i comportamenti emergenti, infatti, nel *quid* che separa questo tipo di sistemi dalla linearità trovano spiegazione tutti quei fenomeni che definiamo complessi e che non sono perfettamente rappresentabili con la matematica lineare.

In particolare la non linearità dei sistemi si presta a spiegare il fenomeno dell'emergenza, cioè il processo grazie al quale, dall'interazione di una moltitudine di “individui”, che possono essere costituiti da esseri umani, come da molecole, come da qualunque altra forma vivente o meno, dà forma ad un'autorganizzazione spontanea, grazie all'interazione reciproca.

Inoltre questi sistemi complessi autorganizzanti sono adattivi, cioè non si limitano a reagire passivamente agli eventi ambientali, ma sono in grado di sviluppare strategie per garantirsi la sopravvivenza all'interno di un ambiente in continua evoluzione, senza la necessità di alcun coordinamento.

In Waldrop (1995), “Così, individui che cercano di soddisfare i propri bisogni materiali si danno inconsciamente una struttura economica basata su infiniti atti di compravendita; e tutto ciò senza una guida o una pianificazione consapevole. (...) Gli studiosi del Santa Fe Institute ritengono di disporre degli strumenti matematici per costruire un tale paradigma di riferimento, attingendo ai fermenti intellettuali degli ultimi vent'anni in campi come le reti neurali, l'ecologia, l'intelligenza artificiale e la teoria del caos. Sono convinti che l'applicazione di queste idee permetterà loro di capire le dinamiche spontanee ed autorganizzanti del mondo in una prospettiva del tutto nuova, con la possibilità di esercitare un impatto immenso sulla conduzione dell'economia, degli affari e della politica. Pensano di porre le basi per la prima alternativa rigorosa al tipo di pensiero lineare, riduzionistico, dominante la scienza fin dai tempi di Newton, e che ha ormai raggiunto i suoi limiti nell'affrontare i problemi del mondo moderno.”

La precedente descrizione di Waldrop, nonostante i limiti dovuti ad una rappresentazione forse eccessivamente divulgativa, riesce tuttavia a far trapelare l'intento principale dello studio della non linearità e della complessità, cioè, come si vedrà in seguito, il tentativo di fornire una rappresentazione della realtà che esula e supera le eccessive semplificazioni, necessarie per una

trattazione matematica tradizionale, ponendo l'accento sull'emergenza dei comportamenti e sull'interazione degli individui che convivono all'interno di uno stesso ambiente.

### **1.2.2 - Comportamenti emergenti.**

Nell'ambito degli studi economici, si è recentemente assistito ad un profondo cambiamento nell'uso degli strumenti d'indagine (Margarita 1992), dai *modelli di struttura*, si è passati all'utilizzo di *modelli comportamentali*, che non considerano l'economia nel suo complesso per analizzarla ed interpretarla attraverso l'utilizzo di tecniche statistico-econometriche, bensì prendono in considerazione i singoli agenti, per valutare il loro comportamento e gli effetti delle loro interazioni reciproche.

L'attenzione degli economisti si è spostata dal cosiddetto *agente rappresentativo*, tipico degli studi dell'economia classica e caratterizzato da razionalità perfetta ed illimitata, ad un individuo, attore del processo economico, soggetto all'influenza dell'ambiente che lo circonda, le cui azioni sono condizionate da quelle di altri individui e, soprattutto, sono caratterizzate da razionalità limitata, informazione imperfetta, difficoltà di scelta e scarsità del tempo necessario a prendere una decisione. Da una visione d'insieme descritta da sistemi di equazioni anche molto complicati, si è dunque passati ad un oggetto d'indagine limitato ai singoli individui eterogenei, le cui risposte semplici a stimoli esterni danno origine a fenomeni complessi, talvolta assolutamente non previsti o, addirittura non prevedibili.

Data questa nuova impostazione, è evidentemente venuta meno l'efficacia dell'uso dei modelli classici, che per mezzo di equazioni dotate di tante variabili quante potevano essere le caratteristiche dell'ambiente valutate a priori, cercavano di descrivere il funzionamento di un sistema, ma non erano in grado di prendere in considerazione sviluppi diversi da quelli prevedibili in partenza. Le scienze sociali, dunque, hanno dovuto ricercare metodi che meglio si adattassero alle nuove esigenze di indagine, pur non prescindendo dall'analisi statistico-econometrica dei risultati raggiunti.

In Testfatsion (1998), si evidenzia come l'impostazione individualista necessiti dell'adozione di un metodo sperimentale, adatto a riprodurre i fenomeni economici con mezzi ed in ambienti artificiali, in grado, in un certo senso, di distaccare l'economista dalla visione classica dell'economia, che vedeva nella storia l'unica fonte di dati per lo studio delle scienze sociali.

Nell'ambito particolare dell'economia, si è individuato nei modelli ad agenti, *ABM*, il mezzo adatto per soddisfare l'analisi dei meccanismi individuali che conducono all'insorgenza di fenomeni complessi e l'eterogeneità degli agenti diviene essenziale, in Terna (1998), " (...) per sperimentare la non linearità degli effetti aggregati dei loro comportamenti (cioè: il tutto che non necessariamente corrisponde alla somma delle parti) in presenza di interazione, con l'emergenza della complessità: è in ciò che consiste l'originalità della sperimentazione artificiale con modelli di simulazione basati su agenti...".



Le perplessità riguardo all'efficacia esplicativa dei modelli ad agenti riguardano, essenzialmente, il grado di complicazione di ciascun agente, la scelta di utilizzare agenti dotati di strutture decisionali e di regole di comportamento semplici, si oppone a quella di usare agenti molto complicati, dotati di regole fortemente strutturate, promossa dall'individualismo metodologico, che proporrebbe di tentare di includere nel singolo agente l'intera complessità del sistema.

In Terna (2000) *"In computer simulation models based upon agents, what is the degree of sophistication that we have to put into the agents? Should we provide them or not with a "mind"? The answer ranges from Axelrod's simplicity principle to the use of full BDI (Beliefs, Intentions, Desires) cognitive agents. To discuss the subject we introduce here three models: one with "no-mind" agents that operate in an unstructured market, the second with "minded" agents assuring some stability to an emerging unstructured market and, finally, the third with no mind agents, that show a sophisticated outcome in a structured market. No generalised results come from this presentation, but many useful doubts"*.

La scelta sembra propendere per l'uso di agenti ragionevolmente semplici e per la ricerca, nell'interazione e nella presenza di strutture *ex-ante*, dei mezzi adatti a riprodurre la complessità presente nel mondo reale. Solo i risultati potranno essere letti alla luce della complessità emersa e trattati con gli adeguati mezzi statistici ed econometrici, al fine di ottenerne una valutazione che esula dalla visione del problema di cui il singolo agente è capace.

### **1.2.3 – Razionalità induttiva.**

La razionalità di tipo deduttivo è in grado di descrivere solo pochi aspetti del comportamento e del ragionamento umani, gli psicologi Schank e Abelson (1977), Rumelhart (1980), Bower e Hilgard (1981), Holland e altri (1986) sono abbastanza concordi nell'affermare che l'uomo, posto di fronte a situazioni e problemi complessi, li semplifica cercando di costruirsi dei modelli interni, effettua cioè deduzioni limitate.

Gli individui, posti di fronte ad un problema molto complicato, lo scompongono in modelli interni via, via più semplici, in modo tale da colmare le proprie lacune, si costituiscono, cioè, delle categorie mentali di portata generale, basandosi su conoscenze ed esperienze particolari.

Grazie ai *feedback* ambientali, le credenze nelle ipotesi correnti di ogni individuo vengono indebolite o rafforzate, mentre alcune vengono scartate e rimpiazzate con delle nuove, quindi, quando non è possibile avere una visione completa di un problema o ne manca una definizione sufficientemente esaustiva, usa modelli semplici per riempire le lacune della propria comprensione, comportandosi in modo induttivo.

Il gioco degli scacchi rappresenta un esempio molto chiaro di ragionamento induttivo, infatti i giocatori riconducono la disposizione delle pedine sulla scacchiera a schemi già visti in precedenza o in partite famose.

Effettuando deduzioni limitate, basate su ipotesi conosciute, ciascun giocatore è in grado di prevedere anche molte mosse successive, via, via che il gioco prosegue e si sviluppa, poi, le ipotesi

si modificano, vengono mantenute solo le configurazioni più plausibili ed i in base ad esse si formulano nuove ipotesi maggiormente coerenti con lo sviluppo del gioco.

#### **1.2.4 – L'esperimento di El Farol: un esempio da Santa Fe.**

Un esempio, che illustra chiaramente l'uso del ragionamento induttivo e dell'emergenza di un fenomeno complesso, generato da agenti dotati di razionalità limitata, è il paper sul bar El Farol di Arthur(1994).

Arthur (1994) propose il problema del bar El Farol come modello di un sistema economico complesso nel quale ogni individuo, appartenente ad una popolazione, deve decidere se recarsi o meno nel locale in occasione della serata settimanale dedicata alla musica irlandese dal vivo.

Dato che lo spazio del bar è limitato, ciascun individuo ritiene che la serata sarà gradevole solo se l'affluenza sarà moderata, cioè se sarà presente meno del 60% delle persone che il locale può contenere: definito in 100 il numero massimo di possibili avventori del bar, ogni individuo deciderà di recarsi se prevede un'affluenza di meno di 60 persone, o deciderà di restare a casa se la previsione è maggiore.

Dal momento che nessun agente possiede un algoritmo di analisi tale da poter prevedere con assoluta certezza quante persone ci saranno nel bar, ciascuno baserà le proprie congetture sull'esperienza passata e, in base ad essa, deciderà se recarsi o meno nel locale.

Quest'impostazione del problema presenta due aspetti fondamentali: in primo luogo, se esistesse un modello ovvio su cui tutti gli agenti potessero fondare le proprie decisioni, sarebbe possibile una soluzione deduttiva ma, dato che ciò non avviene, sono possibili molti modelli diversi di aspettative, dal punto di vista degli agenti non c'è un'aspettativa "corretta", il problema, anzi, è mal definito ed essi sono costretti ad un ragionamento induttivo.

In secondo luogo, non è possibile che più agenti condividano le medesime aspettative, infatti, se tutti credono che pochi andranno nel locale, allora andranno tutti, così come se tutti credono che la maggior parte dei possibili avventori andrà nel bar, allora nessuno andrà; in entrambi i casi le congetture sulla previsione renderanno completamente errata la previsione stessa, quindi le aspettative sono costrette a differenziarsi.

Per costituire un modello di simulazione, che segua le linee dettate dalla descrizione del problema, Arthur (1994) creò degli agenti in grado di formare individualmente ipotesi differenti, in modo tale da poter prevedere le presenze della settimana successiva sulla base delle  $d$  settimane precedenti, dato che nessuna delle regole degli agenti è considerata valida a priori, esse vanno valutate sulla base del successo che hanno dimostrato di avere: ogni agente decide se restare a casa o andare nel locale sulla base dell'ipotesi al momento più accurata, una volta presa la propria decisione, ogni agente apprende il numero delle presenze che si sono effettivamente verificate e modifica l'accuratezza delle sue ipotesi.

E' interessante notare che l'insieme delle ipotesi correnti più accurate, sulla base delle quali gli agenti decidono di agire, determina le presenze nel bar e che le ipotesi attive sono determinate

dalla storia delle presenze, le ipotesi attive sono dunque in grado di formare un'ecologia in grado di evolvere nel tempo, (Holland e altri, 1986).

### 1.2.5 – Analisi del problema.

La simulazione di Arthur adotta la modellizzazione degli agenti descritta in Moss ed Edmonds (1998) ed in Edmonds (1999), in cui ogni agente ha una popolazione di strategie, corrispondenti alle sue possibili scelte, che evolve in base ai propri successi passati.

Ogni singolo agente ha come obiettivo la massimizzazione del proprio punteggio, costituito dall'aver successo andando nel bar quando non è troppo affollato.

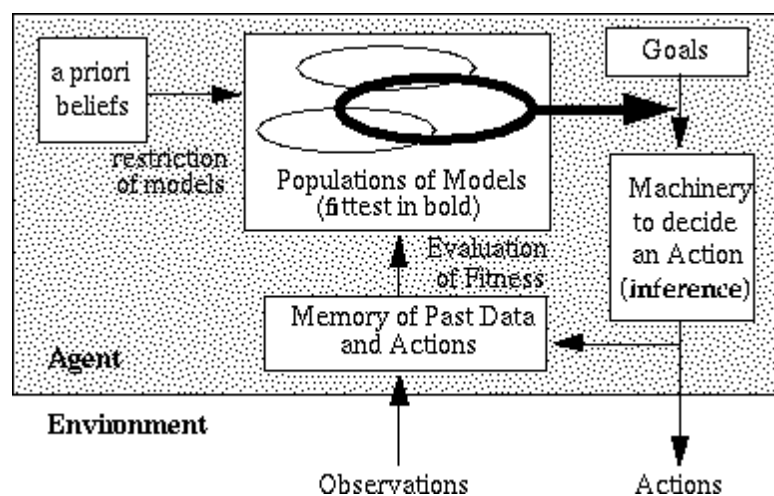
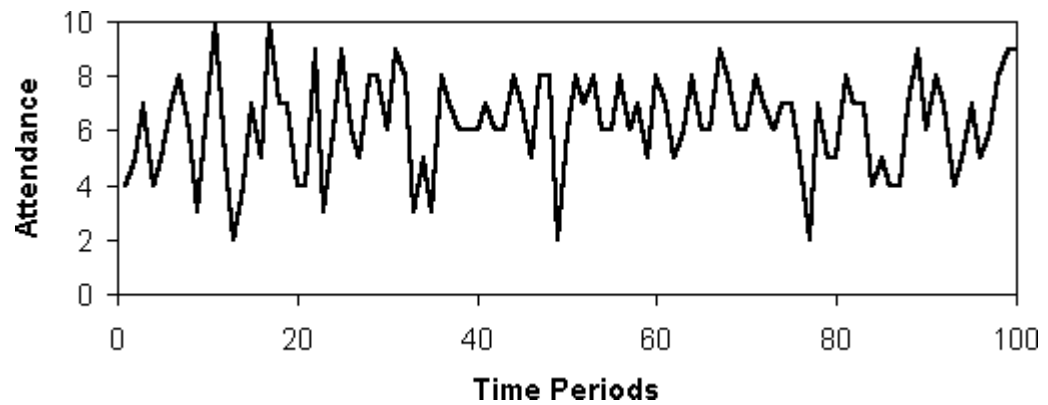


Fig.1.4 – Schema di Funzionamento di ogni agente.

Ogni agente possiede  $k$  ipotesi di partenza alle quali può attingere, ad ogni ciclo, corrispondente ad una settimana, la popolazione delle strategie evolve in base ad un algoritmo genetico.

I risultati dell'esperimento sono interessanti, infatti laddove sono presenti ipotesi basate sull'individuazione di un ciclo, ai cicli viene "ordinato" dalle regole stesse di uscire velocemente, i cicli tendono a non ripetersi con persistenza, dato che, ad esempio, molte persone resteranno a casa se si aspettano che molti vadano perché molti sono andati la settimana precedente.

Notevolmente più interessante, però, è che l'affluenza media converge al 60% delle presenze. Le strategie si auto-organizzano in una situazione di equilibrio, un'ecologia, in cui le strategie attive prevedono per il 40% un'affluenza superiore al 60% e per il 60% prevedono un'affluenza minore di tale soglia.



**Fig.1.5** – Presenze nel bar dopo le prime 100 settimane.

Una spiegazione di questa convergenza può essere determinata dalla presenza di un attrattore naturale del sistema in corrispondenza del 60% delle presenze, infatti, interpretando il problema in un'ottica di teoria dei giochi, risulta che prevedere "più di 60" con probabilità 0,4 e "meno di 60" con probabilità 0,6 è un equilibrio di Nash, che emerge dal comportamento aggregato di tutti gli agenti, senza che vi sia alcun equilibrio previsivo all'interno delle ipotesi scelte.

## BIBLIOGRAFIA CAPITOLO PRIMO.

- ARTHUR B. (1994), *Inductive Reasoning and Bounded Rationality*, American Economic Association Papers, 84, 406-411, Santa Fe Institute, Santa Fe (NM).
- AXTELL R., EPSTEIN J.M. (1996), *Growing Artificial Societies – Social Science from the Bottom Up*, Cambridge MA, MIT Press.
- BAK P. , CHEN K. (1991), *Self-organized criticality*, in Scientific American, NY, vol.264, n.1.
- BATESON G. (1976), *Verso un'Ecologia della Mente*, Milano, Adelphi.
- BOWER G.H., HILGARD E.R., (1981), *Theory of Learning*, Englewood Cliffs NJ, Prentice Hall.
- CASETTI E., (1980), *Equilibrium Population Partitions between Urban and Agricultural Occupations*, Geographical Analysis, 12:1, pp. 47-54.
- DAY R. (1994), *Complex Economic Dynamic, Volume I: an Introduction to Dynamical Systems and Market Mechanism*. Cambridge, Ma, MIT Press.
- EDGAR, PETERS (1991), *Chaos and Order in the Capital Market*, Ny, Wiley and Son.
- EDMONDS B. (1999), *Modelling Bounded Rationality in Agent-Based Simulations using the Evolution of Mental Models*. in BRENNER T., *Computational Techniques for Modelling Learning in Economics*, Kluwer.
- FORRESTER J.W. (1961), *Industrial Dynamics*. Cambridge, MA, Mit Press.
- GLEICK J. (1988), *Chaos: Making a New Science*, NY, Vicking Press.
- HOLLAND J.H. (1975), *Adaptation in Natural Artificial Systems*. Ann Arbor,(MI).
- HOLLAND J..H., HOLYOAK K.J., KEITH J., NISBET R.E., THAGARD P.R. (1986), *Induction*, Cambridge (Mass.), Mit Press.
- HORGAN J. (1995), *From Complexity to Perplexity*, Scientific american, June, 272:6, pp. 104-09.
- HORGAN J. (1997), *The End of Science: Facing the Limits of Knowledge in the Twilight of the Scientific Age*, Paperback ed. New York: Broadway Book.
- KRUGMAN P.R.(1984), *The International Role of Dollar: Theory and Prospect in Exchange Rate Theory and Practice*. BILSON, JOHN F.O., RICHARD C. MARSTON, eds. Chicago: University of Chicago Press, pp.261-78.
- KALDOR N. (1940), *A Model of the Trade Cycle*, Economic Journal. 50 pp.78-92.
- LANGTON C. (1989), *Artificial Life*. Redwood City, Addison-Wesley.
- LE BARON B.(1991), *Empirical Evidence for Non linearities and Chaos Time Series : a Summary of Recent Results*, University of Wisconsin, Social Systems Research Institute, 9117.
- LORENZ H. (1989), *Nonlinear Dynamical Economics and Chaotic Motion*, Berlin, Springer-Verlag.
- MANDELBROT B.B., *The Fractal Geometry of Nature*, ny, W.H. Freeman, 1982.
- MARGARITA S. (1992), *Verso un "robot oeconomicus" algoritmi genetici ed economia*, Sistemi Intelligenti, 3, pp. 421-459, Il Mulino.
- MORIN E. (1997), *Le vie della Complessità*, in *La sfida della Complessità*, Milano, Feltrinelli.
- MOSS S., EDMONDS B. (1998), *Modelling Economic Learning as Modelling. Systems and Cybernetics*, 29, 5-37.  
(Reperibile in <http://www.cpm.mmu.ac.uk/cpmrep03.html>).
- ROSSER J.B.JR (1991), *From Catastrophe to Chaos: A General Theory of Economics Discontinuities*, Boston, Kluwer Academic Publishers.
- ROSSER J.B.JR (1999), *On the Complexities of Complex Economic Dynamics*, in *Journal of Economic Perspectives*, Vol.13, num.4, pp.169-192.
- RUELLE D. (1991), *Chaotic Evolution and Strange Attractors*, Cambridge, Cambridge University Press.
- RUMELHART D. (1980), *Schemata: the Building Blocks of Cognition*, IN SPIRO R., BRUCE B., BREWER W., *Theoretical Issues in Reading Comprehension*, Hillside (NJ), Erlbaum.
- SCHANK R., ABELSON R.P. (1977), *Scripts, Plans and Understanding: an Inquiry into Human Knowledge Structures*, Hillside (NJ), Erlbaum.
- TERNA P. (1998), *Creare mondi artificiali: una nota su sugarscape e due commenti*, Sistemi Intelligenti, 3/98, Bologna, Il Mulino.
- TERNA P. (2000), *The "mind or no mind" dilemma in agents behaving in a market*, in G. Ballot e G. Weisbuch (eds.), *Applications of Simulation to Social Sciences*. Paris, Hermes Science Publications.
- TESTFATSION L. (1998), *The Labor Market: a Simulative Perspective*. (Su: <http://www.econ.iastate.edu/testfatsi/evlab.ps>).
- THOM R. (1975), *Structural Stability and Morphogenesis*.
- THOMPSON J.M.T. (1986), STEWART H.B., *Nonlinear Dynamics and Chaos*, NY, Wiley and Sons.

- VARIAN H.R. (1979), *Catastrophe Theory and the Business Cycle*, Economic inquiry, January, pp.14-28.
- WALDROP M. M. (1995), *Complessità, Uomini ed Idee al confine tra Ordine e Caos*, Torino, Instar Libri.
- ZAHLER R. E SUSSMAN H., (1977), *Claims and Accomplishments of Applied Catastrophe Theory*, Nature 269, pp.759-63.
- ZEEMAN E.C. (1974), *On the Unstable Behavior of the Stock Exchanges*, *Jornal of Mathematical Economics*, 1:1, pp.39-44.

## Capitolo Secondo: Il sistema economico come sistema complesso.

*"Anche se non ci è permesso penetrare negli intimi misteri della natura per apprendere le vere cause dei fenomeni, può nondimeno accadere che una certa ipotesi fittizia sia sufficiente a spiegare molti fenomeni."*  
Leonhard Euler, *Lettere ad una Principessa tedesca*.

La teoria economica, almeno nell'impostazione teorica dominante, è strettamente legata all'andamento dello sviluppo della scienza.

In Lombardi (1981), "L'opera di Walras, uno dei principali esponenti della teoria neoclassica, ha esposto chiaramente il progetto di costruire l'economia sul modello delle scienze naturali: secondo Walras «un'economia politica pura, analogamente a quanto accade nelle scienze fisico-matematiche, deve precedere l'economia politica applicata e come la meccanica e l'idraulica essa deve far uso del metodo razionale, identificato con quello matematico»."

Walras inizia quella che possiamo definire una vera e propria tradizione di ricerca, definita neoclassica, alla quale sono riconducibili almeno due impostazioni teoriche, l'una diretta a realizzare l'idea walrasiana dell'economia come meccanica analitica pura, l'altra volta a perfezionare al massimo il progetto di scienza come costruzione logico-matematica di tipo assiomatico.

L'espressione più compiuta delle due costruzioni teoriche neoclassiche può essere individuata in Samuelson (1973), il quale si poneva alcuni obiettivi espliciti, "dedurre principi generali, che possano servire ad unificare ampi settori della teoria economica di oggi", focalizzare l'attenzione sull'interdipendenza tra le variabili, escludendo rapporti di causalità o determinazione fra di esse, studiare le proprietà delle posizioni di equilibrio attraverso le "variazioni", in un sistema, da una posizione di equilibrio ad

un'altra, senza tenere presente il processo di transizione connesso con l'aggiustamento".

L'idea generale di Samuelson era che, da somiglianze formali nella trattazione teorica di problemi economici, potesse essere dedotta la fondatezza del metodo di generalizzazione per astrazione, da cui successivamente derivare teoremi significativi, formalmente identici in tutti i campi, inoltre, uno dei risultati più rilevanti dell'analisi di Samuelson è la formulazione del principio di corrispondenza fra statica e dinamica, secondo il quale il comportamento di statica comparata di un sistema viene considerato strettamente connesso alle sue "proprietà dinamiche di stabilità".

La seconda costruzione teorica, riconducibile alla tradizione di ricerca neoclassica, è quella che ha portato all'assiomatizzazione dell'equilibrio economico generale di Debreu.

Il punto di svolta, nell'ambito della teorizzazione, ma soprattutto dell'applicazione dei metodi all'analisi economica, è costituito dall'opera di von Neumann, infatti, in Morgenstern, von Neumann (1972), vengono espressi molti dubbi sulla teoria economica e, in particolar modo sul concetto di comportamento razionale, in primo luogo viene messo in discussione l'ipotesi su cui viene fondato il comportamento ottimizzante, proprio delle teorie del consumo basate sul concetto di utilità.

L'identificazione del problema del consumatore con la ricerca di una soluzione di massimo è, in effetti, fondata sulla rappresentazione di un individuo isolato, dotato unicamente di una funzione obiettivo da massimizzare, di cui egli controlla tutte le variabili rilevanti.

In realtà, la situazione di un individuo che partecipa ad un'economia basata sullo scambio è profondamente differente, dato che il singolo agente non controlla tutte le variabili, come vorrebbero le teorie economiche classiche.

I problemi e le difficoltà che nascono nei contesti sociali, richiedono metodi matematici del tutto differenti, rispetto a quelli impiegati nell'economia tradizionale e, soprattutto, occorre focalizzare la propria attenzione sul fatto che le azioni di ciascun soggetto sono influenzate dalle attese circa il comportamento degli altri.

In questo modo nasce la teoria del comportamento razionale in una economia, che si può definire sociale, rappresentata da un gioco con  $n$ -partecipanti, le cui soluzioni sono la risultante delle combinazioni tra le strategie a disposizione degli agenti.

In Morgenstern, von Neumann (1972), "Il problema dell'individuo ha ovviamente molti elementi in comune con un problema di massimo. Ma esso contiene anche elementi essenziali di natura del tutto differente (...) ciascun partecipante allo scambio tenta di massimizzare una funzione (...) di cui egli non controlla tutte le variabili. Questo è certamente non un problema di massimo, ma una peculiare e sconcertante mescolanza di molti e conflittuali problemi di massimo."



Se Morgenster e von Neumann nella loro opera hanno individuato l'importanza dell'interazione fra un numero non definito di agenti che operano all'interno del sistema economico, purtroppo, però, non sono riusciti a superare un altro grande limite dell'economia tradizionale, che è la rappresentazione di un agente dotato di una conoscenza completa, ovvero di un elenco esaustivo degli "stati del mondo" e di tutte le azioni che può compiere in un determinato ambiente.

In Savage (1972), vengono introdotti, seppure in forma embrionale, gli elementi problematici e di critica verso il modello di scelta razionale, che sono la limitatezza della capacità umana di calcolo, secondo la quale un individuo, dunque, non è in grado di conoscere tutti gli stati del mondo, e la necessità per gli agenti di focalizzare l'attenzione su una selezione limitata del contesto decisionale, occorre, cioè che gli agenti si indirizzino a problemi di decisione sufficientemente semplici e ristretti.

I concetti introdotti da Savage sono successivamente ripresi da Simon, che ha sviluppato la teoria della razionalità limitata, partendo dai concetti di incompletezza dell'informazione, di complessità computazionale, di divario fra la complessità del mondo reale e le modeste capacità degli individui nel processare le informazioni.

Secondo il principio di razionalità limitata di Simon, in contesti caratterizzati da incertezza, informazione incompleta e complessità, la procedura decisionale non può essere ottimizzante e l'ottimizzazione approssimata deve costituire il criterio operativo, fondato sulla ricerca di obiettivi "soddisfacenti", connessi ad alternative di scelta provvisorie fino a che non ne vengono trovate di migliori.

In un ambito decisionale caratterizzato da uno spazio delle possibilità molto più ampio di quello ipotizzato dal modello di scelta razionale, all'interno del quale non è possibile definire un algoritmo appropriato per attivare procedure di scelta ottime ed in cui i soggetti ricorrono ad euristiche di ricerca, in modo tale da ridurre lo spazio delle possibilità entro i limiti della trattabilità, per giungere ad un modo di operare "ragionevole", diventano fondamentali le modalità di raccolta e di elaborazione delle informazioni ambientali.

Se l'elaborazione delle informazioni è essenziale per il processo decisionale, altrettanto fondamentale diviene lo studio dei meccanismi cognitivi, perché le regole adottate per decidere dipendono dal modo in cui viene sintetizzata l'informazione, valutata come rilevante ai fini della decisione.

Arthur (1994), cogliendo l'aspetto della complessità all'interno dell'ambiente reale, sviluppa considerazioni interessanti sul sistema di ragionamento induttivo in situazioni decisionali caratterizzate dalla presenza di molti agenti, ciascuno con modelli di credenza ed ipotesi "mentali" sull'ambiente che essi stessi generano per mezzo della loro interazione.

Lo studio dell'economia, e più in generale delle scienze sociali, ha trovato, grazie all'uso del computer, la possibilità di effettuare sperimentazioni vere e proprie, per

mezzo delle quali è in grado di cogliere esattamente questo aspetto della realtà ed ha la possibilità di riprodurre artificialmente le condizioni iniziali di un certo tipo di ambiente e di valutare come da tali condizioni si siano generati i fenomeni, fino ad ora trattati solo come "stato finale" di un processo.

Le simulazioni al computer e, in particolare, i modelli ad agenti adattivi, permettono proprio di analizzare il divenire dei fenomeni e di valutarne l'emergenza, secondo il principio *bottom-up* di Epstein, Axtell.

### 2.1.1 - Mercati reali e mercati artificiali.

Le simulazioni condotte attraverso il computer costituiscono un valido strumento per rappresentare in forma artificiale i fenomeni delle scienze sociali, che difficilmente possono essere trattati con i sistemi di equazioni matematiche fino ad ora adottati anche in campo economico per descrivere il comportamento dei mercati e degli agenti che vi operano.

Tutte le scienze sociali, infatti, sono costituite da sistemi complessi e, proprio per questo motivo, vanno trattate con gli strumenti idonei, che corrispondono agli esperimenti condotti in laboratorio, utilizzati per le scienze naturali.

In Parisi (2001), "Se un sistema complesso è un insieme molto grande di elementi che, interagendo localmente tra loro, producono delle proprietà complessive che non sono deducibili o predicabili, conoscendo anche alla perfezione gli elementi e come interagiscono tra di loro e non sono analizzabili con gli strumenti matematici tradizionali, le simulazioni sono l'unico modo di studiare sperimentalmente i sistemi complessi e, quindi, per studiare i fenomeni umani."

La simulazione, dunque, volta riprodurre in un ambiente artificiale le dinamiche del mondo reale, viene utilizzata per comprendere come dalle interazioni dei singoli agenti possano scaturire comportamenti non predicabili, che potrebbero apparire esogenamente governati.

Le dinamiche di formazione del prezzo all'interno di un mercato, ad esempio, costituiscono un fenomeno che non può essere facilmente studiato con tecniche tradizionali, senza il ricorso ad artifici quali quello del banditore che fissa il prezzo dei beni come ad un'asta o della "*Invisible Hand*" nella quale il fondatore dell'economia moderna Adam Smith individuava un'entità di regolamentazione del mercato completamente esterna al mercato stesso.

La creazione di ambienti artificiali, invece, permette analizzare lo sviluppo del mercato, man mano che questo scopre le proprie regole ed i propri comportamenti, grazie all'interazione dei singoli agenti, che pure non può prescindere dalle strutture del mercato stesso.

In Axtell, Epstein (1996) "*Agents are the "people" of artificial societies. Each agent has internal states and behavioral rules. Some states are fixed for the agent's life, while others change through interaction with other agents or with the external environment. For example, in the model to be described below, an agent's sex, metabolic rate, and vision are fixed for life. However, individual economic preferences, wealth, cultural identity, and health can all change as agents move around*

*and interact. These movements, interactions, changes of state all depend on rules of behavior for the agents and the space."*

Se il comportamento del singolo individuo viene considerato un oggetto di studio trattabile con le tecniche tradizionali, di laboratorio, "i fenomeni sociali sono stati considerati fino ad ora come effetti di cause assolutamente fuori dal controllo dello scienziato e come tali non sottoponibili alle manipolazioni sperimentali."

Attraverso la creazione di mondi e mercati artificiali, però, è possibile analizzare come dal comportamento individuale possano verificarsi effetti aggregati del tutto inspiegabili a priori, che possono essere rappresentati anche dalla comparsa di strutture ed istituzioni sociali che, una volta presenti nel mondo artificiale, costituiscono la struttura dell'ambiente in cui gli agenti operano e ne influenzano il comportamento.

In particolare, le simulazioni condotte mediante l'uso di calcolatori permettono di valutare non solo il trasferirsi delle dinamiche da un punto di vista individuale a quello aggregato, ma permettono di analizzare al tempo stesso gli individui che compongono il sistema e le strutture che nascono dall'interazione degli individui secondo un procedimento *bottom-up*.

L'economia tradizionale ha sempre cercato di spiegare i comportamenti degli operatori economici e l'andamento del mercato mediante metodi riduzionismi, per mezzo dei quali era possibile ricondurre il comportamento di tutti gli individui, agenti in un dato mercato, ad un solo agente "rappresentativo", dotato di caratteristiche quali la perfetta conoscenza del mercato, aspettative razionali, preferenze lineari o funzioni di utilità massimizzanti secondo precise funzioni matematiche.

La realtà, invece, secondo la teoria della "razionalità limitata" di Simon, è popolata di agenti dotati di limitate capacità cognitive e computazionali, e solo dal comportamento collettivo di questo tipo di individui si è generata una struttura tanto imprevedibile e complessa, che si è pensato di poterla interpretare ed analizzare con le metodologie delle scienze naturali e di poterla ricondurre a concetti come quello della razionalità assoluta degli individui.

L'utilità, o meglio la necessità, della simulazione in questo ambito deve, dunque, essere ricondotta al fatto che essa rappresenta l'unico mezzo con cui si possono rappresentare non solo fenomeni quantitativi, ma anche le valutazioni qualitative che gli individui effettuano sul proprio ambiente, prescindendo dalla trattazione puramente matematica.

### 2.1.2 - Complessità nei mercati.

**In Luna, Stefansson (2000) *"Market of goods and services in modern economies are probably among the most complex Dynamic systems which science has tried to analyze and model. [...] Economic and social systems consist of agents which think and react. The elementary particle of matter do not observe, learn and influence each other decisions. Much is therefore lost in the translation from economic behavior to mathematical language of modern economic theory."***

L'idea attuale dei sistemi complessi, è stata introdotta nell'economia senza prestare troppa attenzione ai precedenti contributi al concetto di complessità nella storia del pensiero economico, benché tale concetto fosse familiare allo studio dell'economia.

Molti argomenti di ricerca, come la non linearità e l'autorganizzazione erano stati trattati in precedenza, anche se in forme e contesti differenti, dagli economisti interessati alle complessità dell'economia reale.

Esempi dell'importanza dell'idea di complessità all'interno del pensiero economico moderno possono essere l'interesse di Hayek per le caratteristiche di autorganizzazione dei sistemi complessi, l'enfasi di Keynes sull'instabilità delle posizioni di equilibrio e sul ruolo delle aspettative, la critica di Kaldor al metodo degli equilibri parziali dovuti alla non linearità dei processi economici e le teorie di Frisch sul caos, sulla complessità e sugli effetti della non linearità sui modelli economici.

Gli ultimi studi , in ordine di tempo, focalizzati sulla complessità sono quelli dei ricercatori del Santa Fe Institute; in generale quello che viene definito il "Santa Fe Approach" alla complessità consiste in un tentativo di creare una conoscenza interdisciplinare mediante uno sforzo comune fra il metodo di affrontare lo studio dell'economia "realistico" e quello "artificiale" .

Nonostante l'enorme importanza degli studi del gruppo di lavoro di Santa Fe, sembra ragionevole provare a valutare le eventuali influenze che gli studi precedenti hanno esercitato e, soprattutto, il percorso compiuto dagli studi sulla complessità all'interno del pensiero economico, il particolare per quel che riguarda il punto di vista di Marshall.

Marshall, infatti, riteneva che l'economia fosse una disciplina dotata di una doppia natura: da un lato riteneva l'economia un campo decisamente fruttuoso per le speculazioni intellettuali ed analitiche, sotto un altro punto di vista egli riteneva che l'economia potesse giocare un ruolo fondamentale per il miglioramento delle condizioni dell'umanità.

In Marshall (1898) " economic reasoning should start on methods analogous to those of physical statics, and should gradually become more biological in tone", secondo la natura duale dell'economia marshalliana, le analogie con la biologia sono preferibili a quelle con la meccanica analitica.

La complessità per Marshall era da ricercare non soltanto nell'oggetto di studio, i processi naturali e sociali, ma anche nel metodo utilizzato per trattarli.

Egli pose particolare enfasi sul modo in cui i progressi della conoscenza diventassero complessi in se stessi, osservando che molti fenomeni economici divenivano complessi perché erano valutati secondo prospettive diverse. L'opinione di Marshall sulla complessità del sistema economico potrebbe essere riassunta dall'assunto , in Marshall (1920), "Nature's action is complex: and nothing is gained in

the long run by pretending that is simple, and trying to describe it in a series of elementary propositions”

In realtà, Marshall sviluppo, nella sua analisi, molti elementi che corrispondono al concetto attuale delle teorie della complessità, ad esempio la sua percezione della varietà della natura umana, e l'intuizione che l'economia non può essere valutata con i metodi delle scienze naturali “esatte”, perché il suo corso è in continuo e sottile cambiamento in accordo con le forze che determinano la natura umana.

“L’approccio di Santa Fe”, sviluppa lo studio della complessità coinvolgendo molte discipline diverse, poiché intende considerarla un aspetto di tutti i comportamenti cosiddetti sociali, o collettivi.

Seguendo questa tematica, gli studi della complessità del S.F.I. coinvolgono applicazioni delle dinamiche non lineari all’analisi economica e dei dati, le teorie della dipendenza delle traiettorie dei sistemi dai risultati precedenti, la teoria secondo la quale, una volta che gli individui hanno adottato una strategia la abbandonano difficilmente (*lock-in*), ed infine l’emergenza dei fenomeni e l’evoluzione di sistemi autorganizzati.

Il primo obiettivo di questo tipo di disciplina consiste nello studio di quello che viene definito un sistema complesso adattivo non lineare, caratterizzato dalle seguenti proprietà:

1. I risultati globali sono prodotti dalle interazioni di diversi agenti che operano in anticipazione di altri agenti. La proprietà della diversità produce continua novità nei comportamenti aggregati.
2. Assenza di un controllo globale, perché le interazioni sono prodotte da meccanismi di competizione e coordinamento fra i singoli agenti, senza la necessità di un meccanismo centralizzato che svolga queste funzioni.
3. Un’organizzazione gerarchica trasversale, per la quale esistono diversi livelli all’interno dell’organizzazione generale.
4. Adattamento continuo, gli agenti sono influenzati dall’ambiente ed adattano continuamente il proprio comportamento ai cambiamenti dell’ambiente, secondo un processo di accumulo dell’esperienza.
5. Sorgere continuo di comportamenti e strutture nuovi ed innovativi.
6. Dinamiche al di fuori dagli equilibri, date dallo stato di novità continua, l’economia non opera chiusa all’interno di uno stato di equilibrio, bensì il disequilibrio è una pre-condizione per le dinamiche dei sistemi complessi.

La principale rilevanza dei sistemi complessi, è che essi rappresentano il primo tentativo interdisciplinare di lavorare con strutture teoretiche dalle quali ci si attende che riflettano le condizioni dinamiche reali del mondo.

**Sembra che l'attinenza principale dei sistemi complessi adattivi, per quanto riguarda trattare l'economia, sia la loro capacità di rappresentare una metafora delle dinamiche di base delle strutture economiche del mondo reale.**

### 2.1.3 - Il mercato Azionario come sistema complesso.

I mercati finanziari svolgono un ruolo fondamentale nel processo di allocazione delle risorse. Essi fanno sì che il risparmio possa finanziare l'accumulazione di capitale fisico, il deficit pubblico o l'acquisto di attività finanziarie estere.

Tramite il processo di intermediazione, inoltre, i mercati finanziari rendono possibile la diversificazione verso alcuni tipi di rischio, i mercati azionari ed obbligazionari, infatti, raccolgono fondi dai risparmiatori per distribuirli agli investitori, remunerando i primi in base al grado di rischio sostenuto.

Una funzione fondamentale dei mercati finanziari è quella di guidare gli operatori a muoversi in un contesto rischioso, fornendo loro una forma di protezione, ad un prezzo chiamato *premio per il rischio*, diversificato secondo gli strumenti finanziari offerti ai risparmiatori.

Un'importante caratteristica dei mercati finanziari è nella natura dei beni scambiati, infatti, mentre nella maggior parte dei mercati si scambiano flussi, nei mercati finanziari si scambiano fondi, cioè grandezze di tipo *stock*, costituite da attività finanziarie o reali, che hanno in comune due caratteristiche: sono *standardizzate*, per cui possono facilmente essere scambiate in grandi mercati organizzati e sono durevoli, perché è possibile conservarle ad un costo relativamente basso.

Questi due aspetti fanno anche sì che i mercati delle attività finanziarie siano difficilmente trattabili da un'analisi condotta mediante il paradigma della domanda e dell'offerta, questo significa che anche se, come in ogni altro tipo di mercato, esistono una domanda ed un'offerta, nel mercato finanziario gli agenti economici acquistano attività per investire i propri risparmi e, quindi, non per assolvere una necessità immediata di consumo, ma per rimandare il consumo ad un periodo futuro.

L'agente, dunque, al momento dell'acquisto di un titolo azionario decide in funzione del valore che si attende al momento della vendita di quello stesso titolo e non tanto del valore al momento dell'acquisto.

Dato che le attività rappresentano un modo di trasferire valore nel tempo, i prezzi delle azioni, delle obbligazioni, dei derivati e delle quote di fondi riflettono il valore futuro atteso ed incorporano l'elemento di incertezza, che deriva dal fatto che il valore a cui saranno venduti non è conoscibile al momento dell'acquisto.

Nei mercati finanziari vengono scambiate attività di qualità e di caratteristiche ben conosciute dagli operatori, benché i semplici investitori non siano in grado di trattare volumi sufficienti da essere scambiati direttamente sui mercati, così operano attraverso intermediari, i quali trattano numerosi ordini contemporaneamente o, addirittura, possono "fare il mercato" per conto dei loro clienti, utilizzando i propri portafogli, in questo modo gli intermediari diventano anch'essi detentori

di attività finanziarie e, movimentando grandi volumi di attività, sono in grado di influenzare, in maniera più o meno evidente, l'andamento dei prezzi.

Infatti, avendo la possibilità di investire e disinvestire in tempi molto rapidi, grazie alla contrattazione telematica, e di poter operare su mercati diversi, possono cogliere opportunità di profitto per se stessi e per i loro clienti, che i singoli risparmiatori non potrebbero sfruttare.

Un altro elemento di fondamentale importanza per il mercato azionario, dovuto alla caratteristica del rischio legato al valore futuro atteso dei titoli, è il tipo di previsione che gli operatori effettuano proprio sul prezzo futuro delle azioni e degli altri titoli scambiati.

Gli operatori professionali e sempre più anche i piccoli risparmiatori, infatti, adottano tecniche diverse per prevedere l'andamento del prezzo e degli indici di mercato, tecniche di analisi cosiddetta *fondamentale* o di analisi *tecnica*, le prime considerano soprattutto i dati patrimoniali e finanziari delle imprese, attraverso un'analisi approfondita degli indici aziendali, degli sviluppi del mercato specifico dell'impresa e delle prospettive di crescita dell'azienda stessa, mentre le altre sono maggiormente legate ad un'osservazione di tipo "empirico" delle serie storiche dei prezzi dei singoli titoli e degli indici di borsa.

Gli analisti tecnici si basano soprattutto su una interpretazione visiva e geometrica delle serie storiche dei prezzi, che vengono ricondotte ad alcune figure geometriche giudicate significative dell'andamento del prezzo di un titolo o dell'indice di borsa.

Dall'eterogeneità degli operatori, delle loro convinzioni e delle loro aspettative e dall'interazione di tutti questi tipi di agenti, scaturisce un mercato complesso, il cui andamento, in realtà, è difficilmente prevedibile a priori, poiché spesso si verificano deviazioni da quella che la teoria economica definisce *efficienza*, dal momento che i prezzi non risultano correlati solo con i rendimenti futuri.

Le condizioni di efficienza del mercato, secondo la teoria economica, sono determinate dal fatto che, in primo luogo, i mercati devono raccogliere tutta l'informazione disponibile riguardo al futuro, benché ciò costituisca un costo, in secondo luogo, gli stessi mercati dovrebbero elaborare tale informazione in modo tale da effettuare valutazioni corrette del futuro e del rischio e, se i mercati sono efficienti, nessun operatore deve preoccuparsi di reperire informazioni aggiuntive, poiché essi le rivelano attraverso i prezzi, rendendo inutile qualunque tipo di previsione diversa da quelle dell'analisi fondamentale; in Friedman (1953), si sostiene che la speculazione non può essere destabilizzante, perché gli operatori che spingono il prezzo di un titolo lontano dal suo valore «fondamentale» sono costretti ad uscire dal mercato.

Ciò che è veramente interessante è valutare come e perché i mercati tendono a discostarsi da una situazione di efficienza.

Una delle possibili risposte è che la mancanza di efficienza sia dovuta proprio all'esistenza, all'interno di un mercato, di operatori non professionisti, che basano le loro previsioni su elementi irrazionali o che si basano su entusiasmi passeggeri, dando origine, grazie all'interazione collettiva, alla nascita di bolle speculative o di persistenti deviazioni dei prezzi dai valori fondamentali.

Un esempio di questo tipo di comportamento, si è avuto nel prezzo delle azioni ordinarie della Tiscali nel corso del 1999-2000, infatti, sull'onda dell'entusiasmo per i titoli tecnologici, determinato dall'andamento

continuamente ascendente dell'indice Nasdaq negli USA, tali azioni hanno raggiunto livelli di prezzo molto superiori al loro valore fondamentale, per poi ridimensionarsi dopo il parziale crollo del Nasdaq nella primavera del 2000.

Il valore fondamentale di un'azione è determinato dal valore attuale di una serie infinita di dividendi  $d$  scontati al tasso  $r$ .

$$\hat{q} = d/r \quad \text{e} \quad Dq/q = 0$$

I prezzi di un titolo che si allontanano da questo valore, senza alcuna apparente giustificazione, sono «bolle», che si autoalimentano, infatti, finché gli operatori credono che la bolla continuerà ad espandersi, saranno disposti a detenere il titolo o ad acquistarlo e ciò è sufficiente per alimentare la bolla. Le aspettative, cioè, si convalidano *ex post*.

D'altra parte, affinché la bolla continui ad espandersi deve prevalere l'aspettativa che il prezzo cresca all'infinito, altrimenti, se cioè è noto il momento in cui il prezzo si stabilizzerà, la bolla si sgonfia immediatamente, perché un'ipotesi di stabilizzazione del prezzo implica che questo, nelle aspettative degli operatori, sia pari al suo valore fondamentale.

La storia fornisce molti esempi di bolle speculative, in Gaber (1990) viene descritta la «mania dei tulipani», verificatasi nel 1637, come una delle bolle più famose, durante la quale si ebbe una crescita spropositata del prezzo dei bulbi di tulipano (3000 punti percentuali in poco più di due mesi), seguita da un crollo improvviso.

Esempi più recenti e più pertinenti al mercato azionario sono le crescite dei prezzi delle azioni che hanno preceduto i crolli storici di Wall Street e dei mercati mondiali nel 1929, nel 1987 e nel 1989.

Un'altra possibile spiegazione delle bolle è che non tutti gli operatori siano razionali, come ipotizzato dai modelli economici tradizionali, ma che solo una parte di essi sia informata e riesca a valutare il reale valore dei titoli, mentre i rimanenti siano *noise traders*, cioè agenti irrazionali o male informati, che giungono continuamente sul mercato per sostituire quelli che sono stati costretti ad uscirne.

Pur ipotizzando che sul mercato operino dei professionisti, che adottano un comportamento razionale e perfettamente efficiente, la presenza dei *noise traders*, potrebbe giustificare la divergenza per un lungo periodo dei corsi azionari dai loro valori fondamentali.

Attraverso i modelli di simulazione *ASM* e, in particolare *SUM* che vedremo dettagliatamente nei capitoli 5 e 6, si intende analizzare come la natura degli agenti che operano in un mercato di borsa, possa influire sui meccanismi di formazione del prezzo e sui livelli del prezzo stesso, e di come i processi di interazione degli agenti possano essere significativi, poiché riescono a cogliere l'aspetto complesso del mercato che i modelli tradizionali non sono in grado di rappresentare.

#### **2.1.4 - ASM, il modello del SFI: un esempio.**



L'*Artificial Stock Market (ASM)*, è un modello creato da Arthur, Holland, LeBaron, Tayler e Palmer, per rappresentare un mercato finanziario basato su agenti, dotati di una capacità di apprendimento basata su *classifier system*.

Nel mercato reale, gli agenti cercano di accrescere il proprio patrimonio grazie a regole semplici, spesso difficilmente rappresentabili secondo una qualsiasi formalizzazione matematica, basate soprattutto sulle esperienze passate e suscettibili di continue variazioni per adeguare il proprio comportamento all'andamento del mercato.

Nel mercato artificiale *ASM*, invece, gli agenti sono in grado di effettuare previsioni sul mercato basandosi su aspettative razionali, elaborate grazie ad una selezione delle informazioni disponibili nell'ambiente.

Gli agenti del modello *ASM* hanno a loro disposizione due tipologie di attività finanziarie, che si differenziano fra loro per il grado di rischio e per il tipo di remunerazione che offrono, rispettivamente un tasso di interesse costante ed un dividendo stocastico determinato secondo un procedimento autoregressivo.

Il modello si configura come un modello neoclassico: gli agenti sono avversi al rischio con funzione di utilità di tipo CARA (con avversione assoluta al rischio costante) ed hanno una funzione di domanda standard per l'attività rischiosa, basata sulla previsione del prezzo e del dividendo del periodo successivo, inoltre, una volta determinate le previsioni, calcolano la domanda per il titolo rischioso e la 'sottopongono' ad un banditore (*specialist*), che determina il prezzo di equilibrio (in modo da uguagliare la domanda aggregata all'offerta fissata), le risorse monetarie non investite nell'attività rischiosa sono impiegate nell'altro titolo disponibile "*risk free*" (*cash*, nella simulazione).

L'elemento che distingue il modello *ASM* da un modello economico tradizionale, essenzialmente descrittivo, è l'uso di agenti in grado di formulare previsioni dinamiche ed eterogenee, ossia modificabili in base a regole che evolvono, mediante un algoritmo genetico, in modo autonomo per ogni agente.

Le previsioni, che possono essere diverse per ogni agente, quindi, determinano a loro volta le funzioni di domanda degli agenti, anch'esse eterogenee, poiché dipendono dal prezzo e dal dividendo previsti.

Ogni agente deve formulare previsioni sui prezzi e sui dividendi futuri dell'attività rischiosa, utilizzando regole selezionate grazie ad un *classifier system*, che agisce per mezzo di un meccanismo di condizione-azione. Le regole di classificazione (*classifier rules*) descrivono, mediante un sistema di codificazione binaria, le condizioni del mercato e, se la condizione corrisponde ad una regola, allora quella regola è potenzialmente utilizzabile per prevedere il prezzo e il dividendo del periodo successivo. In uno stesso periodo più regole di previsione possono corrispondere allo stato del mercato, ed essere, quindi, potenzialmente attivate, dando così all'agente una maggiore possibilità di scelta.

**In un momento successivo, le regole vengono monitorate secondo l'accuratezza della previsione che hanno generato e solo quelle migliori vengono attivate e utilizzate**

**per creare, in un processo di apprendimento periodico, nuovi set di regole, mediante l'utilizzo di un algoritmo genetico.**

Gli agenti del modello *ASM* hanno a loro disposizione sia regole di analisi tecnica, sia regole di analisi fondamentale, benché i risultati dell'attivazione o meno di queste regole non fornisca al modello di aspettative razionali alcun elemento addizionale, infatti, la previsione del prezzo e del dividendo futuri è basata esclusivamente su quelli correnti.

Le regole di classificazione sono costituite da due parti, la prima è una stringa di bit, caratterizzata dai simboli 1, 0, #, in cui lo zero e l'uno descrivono la presenza o meno di uno stato, mentre il carattere # indica uno stato di indifferenza, la seconda parte della regola, invece, ha il compito di convertire il set di bit in una previsione del prezzo e del dividendo.

Una volta che una regola viene attivata ed applicata, l'agente sostituisce i parametri di previsione nella funzione di domanda dell'attività rischiosa.

Date le domande di tutti gli agenti, il prezzo del titolo rischioso viene facilmente ricavato uguagliando la domanda totale all'offerta fissata. Una volta definito l'acquisto dei titoli, gli agenti aggiornano l'accuratezza delle regole di previsione utilizzate secondo una media ponderata.

Gli operatori in fase di contrattazione sono soggetti ad alcune limitazioni: possono contrattare un massimo di 10 titoli in ogni periodo, non possono indebitarsi, e non possono superare un limite di vendite allo scoperto fissato in 5 titoli.

Gli agenti, nel corso dello svolgimento della simulazione, cambiano il proprio comportamento modificando il set di regole di previsione a disposizione, eliminando il 10% delle regole che hanno ottenuto le peggiori prestazioni e sostituendole da nuove regole ottenute dall'evoluzione dell'algoritmo genetico, attraverso la mutazione e l'incrocio del restante 90% di regole sopravvissute.

Tale meccanismo viene attivato in corrispondenza di un ciclo diverso per ogni agente, pur mantenendo una regolarità per il singolo agente, per non far coincidere l'apprendimento di tutti gli agenti nello stesso periodo. La scelta delle regole da rimpiazzare e di quelle che hanno ottenuto le migliori prestazioni, da utilizzare per generare nuove regole, avviene sulla base del numero di caratteri di indifferenza, cioè vengono scartate, a parità di prestazione, le regole che contengono un maggior numero di simboli *don't care* (#).

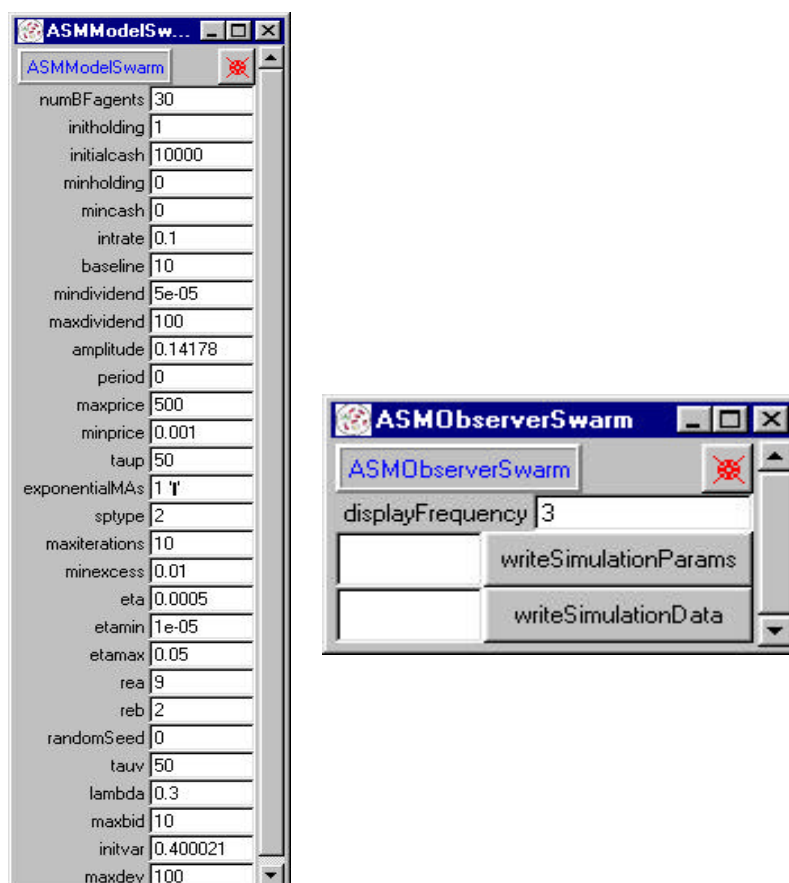
L'algoritmo genetico opera attraverso il meccanismo dell'incrocio con una probabilità dello 0.3, e della mutazione con una probabilità dello 0.01.; attraverso l'incrocio, le nuove regole vengono generate combinando il patrimonio genetico di due regole genitrici, scelte fra quelle superstiti della popolazione iniziale, mentre, attraverso la mutazione, viene modificato un carattere all'interno della stringa della regola genitrice.

Al principio della simulazione, le regole sono inizializzate casualmente, i bit della stringa degli stati vengono poste uguali a # con probabilità 0.9 e uguali a 1 o 0 con probabilità 0.1, le regole, quindi sono il più generali possibile per non influenzare il risultato e non legarlo in nessun modo alle condizioni iniziali. Ogni agente viene, infine, dotato di un'unità dell'attività rischiosa e di 10000 unità di moneta.

Il modello di mercato finanziario artificiale *ASM* è stato realizzato con l'utilizzo di *Swarm* ed ha, quindi, il vantaggio di essere facilmente riproducibile.

Nella finestra dell'*ASMModelSwarm* è possibile definire e modificare le variabili relative ad alcune caratteristiche strutturali del modello, come il numero degli agenti, la dotazione iniziale delle attività finanziarie ad alto rischio, il limite di vendite allo scoperto ed il limite di indebitamento, il tasso di interesse sull'impiego esente da rischio, i minimi e massimi per dividendi e prezzi, il valore centrale attorno al quale viene calcolato il dividendo e l'ampiezza delle deviazioni da tale valore il periodo di autocorrelazione del processo, se utilizzare una media mobile pesata esponenzialmente o, semplicemente, una media mobile di  $n$  periodi per la definizione dell'adeguatezza delle regole, il coefficiente di avversione al rischio, il parametro che definisce l'orizzonte temporale preso in considerazione dall'agente, il tipo di banditore e i parametri da utilizzare nel calcolo del prezzo di equilibrio, il seme d'innescio dei numeri casuali, l'offerta massima di attività rischiose, la varianza iniziale e la massima deviazione di una previsione nella stima della varianza.

Dalla finestra dell'*ASMObserverSwarm* è invece possibile impostare tutti i parametri del grafico e di salvare su un file i parametri della simulazione.



**Fig.2.1** – Model ed Observer del modello *Swarm ASM*.

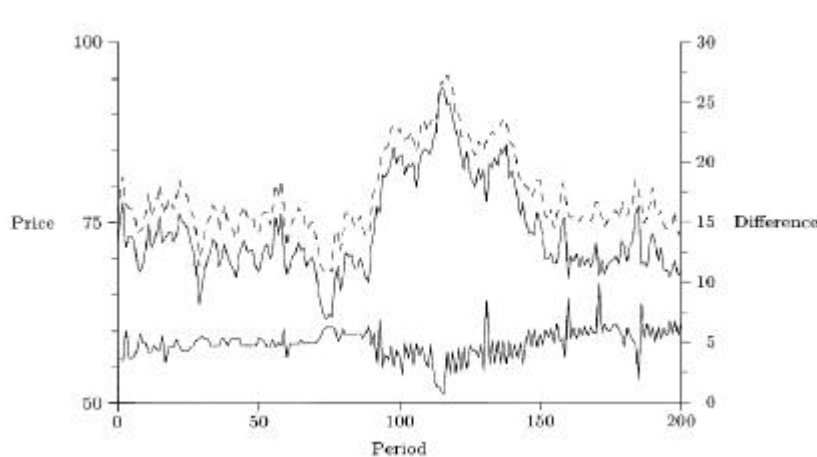
In LeBaron (2000), è stata compiuta un'analisi dei risultati di venticinque diverse simulazioni del modello ASM, ognuna con un differente innesco dei numeri casuali, e con due differenti valori della frequenza di apprendimento ( $k=250$  e  $k=1000$ ).

Ogni simulazione è stata fatta lavorare per 250.000 periodi in fase di apprendimento moderato e sono state analizzate le serie storiche dei successivi 10.000 periodi.

**Di seguito è riportata la dinamica dei prezzi in una delle 25 simulazioni eseguite con frequenza di apprendimento pari a 250.**

La linea continua descrive la dinamica del prezzo effettivo, quella tratteggiata il prezzo teorico nell'equilibrio ad aspettative razionali e la linea in basso nel grafico la differenza tra i due prezzi.

Le serie storiche ottenute dalla simulazione, evidenziano un insieme di periodi nei quali la differenza tra il prezzo teorico di equilibrio ad aspettative razionali (REE) e il prezzo che si realizza effettivamente non si allontana molto da un valore costante, e periodi, invece, in cui questa differenza fluttua ampiamente. Tale fenomeno risulta una caratteristica comune per molte delle simulazioni eseguite.



**Fig.2.2** –Grafico del prezzo in un modello con apprendimento frequente

Il mercato, in pratica, attraversa delle fasi nelle quali la valutazione delle attività finanziarie da parte degli agenti, avviene in coerenza con l'equilibrio ad aspettative razionali e fasi, invece, nelle quali tale valutazione segue regole differenti.

Attraverso un'analisi econometria delle serie storiche dei prezzi, generate dal modello, è stato possibile individuare nella frequenza dell'apprendimento un elemento di differenza molto forte, infatti, se l'apprendimento viene fatto solo dopo molti cicli di *run*, gli agenti del modello riproducono una situazione molto simile a quella di un equilibrio in caso di aspettative razionali, siamo, quindi, molto prossimi ad uno stato

che l'economia classica definirebbe "perfetto", mentre, se l'apprendimento è molto frequente, le serie storiche dei prezzi si allontanano decisamente da questa situazione e riproducono in maniera abbastanza fedele le oscillazioni di prezzo che si presentano in un mercato reale.

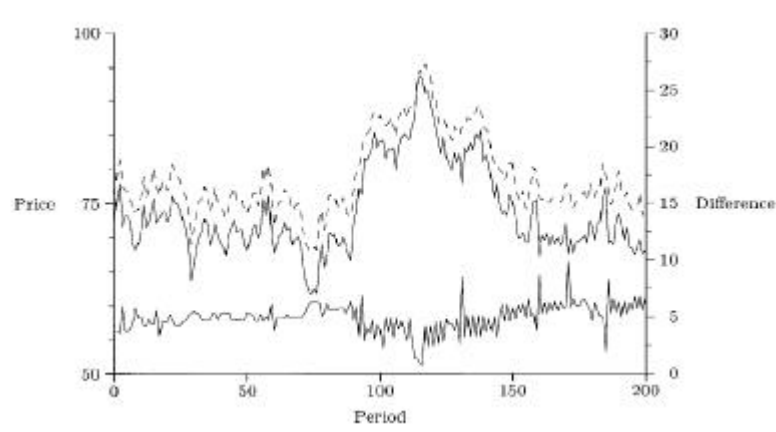


Fig. 2.3 – Grafico del prezzo con apprendimento poco frequente

La conclusione alla quale giunge LeBaron è che, quando l'algoritmo genetico viene applicato con una frequenza relativamente bassa, le regole di previsione evolvono lentamente, i prezzi sono più stabili e, conseguentemente, lo sviluppo delle regole è più semplice, quindi, è più complicato individuare l'uso da parte degli agenti di informazioni assimilabili all'analisi tecnica ed i volumi scambiati sono modesti. Il mercato sembra in grado di raggiungere l'equilibrio ad aspettative razionali.

Quando, invece, l'algoritmo genetico viene chiamato ad evolvere le regole più frequentemente, la varianza delle serie storiche dei prezzi aumenta, il volume delle attività scambiate cresce, conseguentemente lo sviluppo delle regole è più complesso e gli agenti fanno un maggior uso di informazioni di scarso fondamento economico, derivanti dall'osservazione del mercato.

Nel secondo caso, inoltre, la ricchezza accumulata dagli agenti risulta mediamente inferiore a quella ottenuta precedentemente, il mercato, in pratica, non riesce a "stabilizzarsi" e si colloca in una condizione di sub-ottimalità.

La lettura di questi risultati indica che, se gli agenti operano in un mercato in continua evoluzione, saranno portati a rivedere periodicamente le proprie previsioni, tendendo a considerare anche elementi irrazionali, caratteristici del comportamento individuale, giungendo ad una perdita generale di profitto.

Se, invece, gli agenti vengono dotati di un orizzonte temporale sufficientemente ampio, i comportamenti individuali tendono ad annullarsi e prevalgono comportamenti razionali, basati su

analisi di tipo fondamentale, che conducono ad una situazione molto simile a quella dell'equilibrio ad aspettative razionali.

#### **BIBLIOGRAFIA CAPITOLO SECONDO.**

- ARTHUR W.B. (1994), *Inductive Reasoning and Bounded Rationality*, in *American Economic Review*, vol.84, n.2, pp. 406-411.
- FRIEDMAN M. (1953), *The Effect of Full Employment Policy on the Economic Stability: A Formal Analysis*, in *Essays in Positive Economics*, Chicago, University of Chicago Press.
- GABER P. (1953), *The Dollar Bubble*, in *Economics of the Dollar Cycle*, a cura di Gerlach S.e Petri P., Cambridge, Ma, MIT Press.
- JOHNSON P. (2000), *ASM documentation*, reperibile all'url : <http://artstkmkt.sourceforge.net>
- LEBARON B. (2000), *Agent-Based computational finance: suggested readings and early research*, in *Journal of Economic Dynamics and Control* 24, pp 679-702
- LEBARON B., ARTHUR W.B., PALMER R., (1999), *Time series properties of an artificial stock market*, in *Journal of Economic Dynamics and Control* 23, pp 1487-1516.
- LOMBARDI M. (1981), *Keynes, ovvero dell'economia non euclidea*, in *Studi Economici*, 15, pp.76-114.
- LUNA F., STEFANSSON B. (2000), *Economic Simulations in Swarm: Agent -Based Modelling and Object Oriented Programming*, Boston, Mass, Kluwer Academic Publisher.
- MARSHALL A. (1898), *Distribution and Exchange*, *Econ.J.*, 8 (1), 37-59.
- MARSHALL A. (1920), *Principles of Economics*, London, MacMillan Press.
- PARISI D. (2001), *Simulazioni. La realtà rifatta nel Computer*, Bologna, Il Mulino.
- SAMUELSON P.A. (1973), *Fondamenti di analisi Economica*, Milano, Il Saggiatore.
- SAVAGE L.J. (1972), [1954], *The Foundation of Statistic*, N.Y., Dover Publications.
- SIMON H.A. (1985), *Le teorie della Razionalità Limitata*, in *Id. Causalità, Razionalità ed Organizzazione*, Bologna, Il Mulino.

## Capitolo Terzo: Intelligenza Artificiale.

“ Ragionare non è nient'altro che calcolare.”

Thomas Hobbes, *The Leviathan*.

Nella considerazione comune, l'intelligenza umana rientra tra quei fenomeni che trascendono il mondo fisico e le sue leggi. Le potenzialità della mente ed il suo funzionamento non sono ancora del tutto spiegati, ma, a livello filosofico il problema ha sempre suscitato molto interesse e le teorie al riguardo si sono sviluppate in due direzioni. L'una considerava la mente irriducibilmente separata dalla materia, l'altra, invece, cercava di conciliare la teoria della mente con le leggi della realtà materiale e giungeva alla conclusione che il pensiero fosse, in ultima analisi, una forma di calcolo che la mente effettua manipolando qualche tipo di simboli o idee.

Questa idea, esposta in modo molto semplificato, ha attraversato la tradizione culturale occidentale fino alla metà del nostro secolo, quando si è incontrata con la nascente scienza dei calcolatori.

Se, infatti, è corretto identificare il pensiero con una qualche forma di calcolo, non esiste nessuna ragione per cui un computer non possa avere una “mente intelligente” assimilabile a quella umana.

A partire dagli anni '50, quindi, si sviluppa sempre più interesse per una nuova disciplina, l'Intelligenza Artificiale (IA), il cui obiettivo può essere definito *l'indagine sui meccanismi soggiacenti alle facoltà cognitive degli esseri umani* (come il linguaggio, il ragionamento, la capacità di risolvere problemi complessi, la percezione) *e la loro riproduzione mediante il computer.*

A seconda dell'intensità di significato attribuito alla parola *riproduzione* precedentemente usata si distinguono due tipi di IA: l'IA *Forte* e l'IA *Debole*.

L'IA Forte coinvolge diverse discipline, quali biologia, psicologia, biochimica ed anatomia e si propone di dotare i computer di una vera e propria intelligenza, basata su processi cognitivi e mentali analoghi a quelli umani, non essendone in alcun modo distinguibili, come scrive John Haugeland (1988): "L'IA vuole solo originali: macchine dotate di mente in senso pieno e letterale".

L'IA Debole si colloca in un contesto più ristretto, cioè quello dell'informatica, e ritiene che un computer, opportunamente programmato, possa simulare i processi cognitivi umani, senza fare assunzioni sul fatto che il modo in cui lo fa coincida ad un qualche livello con i processi mentali umani. L'IA Debole, dunque, si rivela uno strumento flessibile e facilmente valutabile per quanto riguarda la bontà dei risultati.

### 3.1 – Cognitivismo e Connessionismo.

La *scienza cognitiva* iniziò a svilupparsi nella seconda metà degli anni '50 grazie alla convergenza di diverse discipline rivolte allo studio dei sistemi intelligenti. Il *cognitivismo*, infatti, riunisce fondamenti di psicologia cognitiva, di neuroscienze, di linguistica e di filosofia del linguaggio, assimilando le innovazioni concettuali e tecnologiche offerte dall'informatica, grazie alle teorie di IA.

La scienza cognitiva può essere interpretata come l'erede dell'IA Forte, soprattutto per la convinzione che un sistema intelligente sia fundamentalmente un sistema di elaborazione delle informazioni, che interagisce con un ambiente complesso e che prescinde dalla struttura biologica di un cervello umano.

Adottando i presupposti della psicologia cognitiva, infatti, il *cognitivismo* concepisce la mente come un sistema che elabora informazioni provenienti dall'ambiente e si propone l'obiettivo di coglierne l'architettura ed il meccanismo di funzionamento per dedurre regole il più possibile generalizzate, per riprodurre modelli intelligenti al di fuori del cervello.

Questo filone teorico ha suscitato perplessità dovute, in primo luogo, al fatto che, al livello attuale degli studi sul cervello, non sembra plausibile ravvisare meccanismi funzionalmente equivalenti a quelli di una macchina formale automatica a stati discreti e sequenziali.

Infatti, per quanto si è potuto finora verificare, il cervello non funziona in base a processi sequenziali, bensì opera tramite milioni di processi paralleli, a loro volta definiti da continue variazioni di stato. In secondo luogo, nonostante gli sforzi delle discipline afferenti all'IA, esistono capacità cognitive legate alla percezione, che non sembrano semplificabili o riducibili a processi computazionali di tipo simbolico.

Le precedenti considerazioni sono alla base del superamento del *cognitivismo*, alla fine degli anni '70, in favore di un nuovo filone teorico, il *connessionismo*, termine



coniato per mascherare i concetti di rete neurale e di perceptrone che Minsky (1969) aveva ritenuto definitivamente superati in quanto, secondo il suo parere, rappresentavano dei sistemi matematici senza soluzione.

Il fondamento di questa nuova teoria, stravolgendo le tesi dell'IA classica, si basa sul principio che, per spiegare e riprodurre l'intelligenza umana e le sue facoltà cognitive, occorre riprodurre anche il funzionamento strutturale. Occorre, dunque, emulare il funzionamento fisiologico del cervello studiando il tipo di legame che si instaura fra le cellule cerebrali e, per fare ciò, è necessario attingere agli studi sin qui effettuati nell'ambito delle neuroscienze.

Con connessionismo si indicano sistemi che cercano di imitare le principali caratteristiche di una struttura cerebrale, cioè di una struttura costituita da nodi (neuroni), fra cui esistono legami caratterizzati da intensità diverse e, soprattutto, variabili, in grado di produrre degli output non predeterminati.

In Terna (1995), " Il connessionismo è attualmente impiegato per identificare sistemi - ad esempio le RNA - paralleli, subsimbolici, autorganizzanti, nonché robusti verso gli errori e ridondanti".

Un sistema connessionista, dunque, è caratterizzato da una struttura parallela, nella quale i processi computazionali vengono effettuati tramite una ripartizione fra i vari nodi; il risultato non dipende strettamente da nessun nodo, ma tutti, simultaneamente, contribuiscono all'elaborazione dell'output finale.

Analogamente, all'interno di una RNA, non esiste un nodo specifico che, da solo, detenga la conoscenza ed abbia le soluzioni del sistema, esistono, invece, dei legami fra i nodi, dotati di pesi variabili ed in grado di attivare o non attivare i neuroni, al fine di contribuire al raggiungimento diretto o indiretto del risultato stesso.

Un sistema connessionista, inoltre, è autorganizzato: una rete neurale, attraverso meccanismi di correzione degli errori (come la *backpropagation*), adatta automaticamente i pesi dei suoi legami giungendo, attraverso molti cicli successivi di apprendimento, ad ottenere un errore nullo e, dunque, un'efficace autodeterminazione dei parametri della funzione stessa.

Infine, la robustezza dell'errore e la ridondanza sono strettamente legate al parallelismo ed alla distribuzione di elaborazione dei processi, infatti la disattivazione di un nodo, in un sistema dotato di un numero di nodi sufficientemente alto e di una struttura di pesi ben organizzata, non produce effetti estremamente rilevanti sull'errore, poiché è possibile supplire al nodo mancante rielaborando i pesi e ridistribuendo ulteriormente la conoscenza.

### 3.1.1 – Il test di Turing.

Uno dei primi problemi incontrato negli studi sull'IA è stato quello di avere la certezza dell'intelligenza di un computer: come si può capire se un computer è effettivamente intelligente? Una possibile soluzione potrebbe consistere nel trovare una definizione univoca di intelligenza, ma è molto difficile tracciare dei confini precisi per una materia che presenta ancora troppi lati oscuri, anche rimanendo nell'ambito più familiare dell'intelligenza naturale.

Il matematico inglese Alan Turing in *Macchine calcolatrici ed intelligenza (1950)* inizia con questa frase: "Mi propongo di considerare la domanda: Possono le macchine pensare?" e per rispondere al quesito propone un metodo noto come Test di Turing.

Si immagini che ci siano un uomo ed una macchina in una stanza e che in un'altra stanza vi sia un'altra persona che li interroga attraverso una telescrivente.

Non avendo, a priori, alcun modo di distinguere l'essere umano dal computer, se, alla fine dell'interrogazione, l'interrogante non riesce a distinguere le risposte dell'uomo da quelle del computer, allora quest'ultimo ha superato il test di intelligenza di Turing.

Le domande poste, chiaramente, devono essere di carattere molto generale e di un gran numero di argomenti. Se infatti si ponessero domande di carattere matematico molto complicate, il computer risponderebbe immediatamente, mentre l'uomo impiegerebbe più tempo, quindi sarebbero immediatamente individuabili o, altrimenti sarebbe facile ingannare il test con semplici trucchi di programmazione, semplicemente ritardando la risposta della macchina.

Secondo Turing, opportunamente studiato, il test sarebbe in grado di sostituire la domanda: Può una macchina pensare? E sarebbe, altresì in grado di fornirne un'esauriente risposta.

### **3.1.2– Due esempi di elaborazioni sulla base del test di Turing: I caso di Eliza il dottore e Parry il paranoico.**

Una famosa elaborazione del Test di Turing è costituita dal programma Eliza, sviluppato nel 1964 da Joseph Weizenbaum, giovane ricercatore del MIT. Eliza era un programma con più copioni, il più famoso dei quali simulava, o meglio recitava la parte di uno psicoterapeuta di scuola rogersiana. Gli psicoterapeuti di questa scuola, durante le sedute, cercano di intervenire il meno possibile e, quando lo fanno, spesso si limitano a riprendere le affermazioni del paziente.

Il modello linguistico, quindi, era abbastanza semplice da simulare. Una volta ultimato il programma, Weizenbaum lo installò su un computer del suo istituto e lo fece provare a diverse

*persone. Eliza ebbe uno straordinario successo: la gente che conversava con il programma credeva di parlare con uno psicoterapeuta e provava addirittura sollievo dalla seduta.*

*Uno psichiatra, Kenneth Colby, scrisse che entro pochi anni programmi come Eliza sarebbero stati pronti per l'uso clinico. Poco dopo Colby realizzò un programma simile ad Eliza, Parry, che simulava il comportamento di un paranoico e che ingannò molti psichiatri i quali, dopo averlo interrogato con una telescrivente, dichiararono di aver avuto in seduta un vero paranoico.*

Sembrava, dunque, che questi due programmi avessero superato il test di Turing. In realtà, né Eliza, né Parry erano dotati di intelligenza, in entrambi i casi si trattava di un insieme di trucchi di programmazione che facevano affidamento sulla credulità degli interlocutori.

A ben vedere, poi, nessuno dei due avrebbe mai superato il test di Turing nella forma in cui Turing stesso lo aveva immaginato, perché l'ambito della conversazione era troppo ristretto per simulare una vera e propria intelligenza.

### **3.1.3 – Simboli e significati.**

Nell'ambito dei fondamenti teorici dell'IA emerge che la capacità di elaborare simboli è alla base del comportamento intelligente ed il più completo esempio di elaborazione simbolica degli esseri umani è sicuramente il linguaggio verbale.

Linguaggio ed intelligenza sono fenomeni strettamente interconnessi, come emerge anche dal citato test di Turing, secondo il quale un eventuale computer intelligente dovrebbe saper comunicare mediante il linguaggio naturale. Le ricerche sull'elaborazione di tale linguaggio (*natural language processing*), avendo assunto un ruolo determinante, sono divenute oggetto di una disciplina autonoma, la linguistica computazionale.

Chomsky (1979) introduce il concetto di competenza linguistica, definendolo come quell'insieme di conoscenze implicite, presenti nella mente, che consentono ad ogni essere umano di produrre frasi e discorsi nella sua lingua.

La competenza linguistica, secondo Chomsky, si distingue fra competenza fonologica, competenza sintattica e competenza semantica, che è la capacità di assegnare alle parole un significato o di estrarlo da una frase.

La competenza semantica è sicuramente l'aspetto linguistico che riguarda più da vicino il concetto di intelligenza e che ci porta, attraverso le teorie del filosofo John R. Searle, alla fondamentale distinzione fra simbolo e significato.

In un articolo del 1980, "Menti, Cervelli e Programmi", Searle dichiara che se una macchina può certamente manipolare simboli formali e restituire altrettanti simboli formali, essa sarà solo sempre

dotata di una sintassi, utilizzando una terminologia linguistica e collegandoci alla teoria chomskiana.

Quello che la macchina non avrà mai è una semantica, cioè la capacità di legare i simboli ad un significato. Searle sostiene che, dato che le manipolazioni di simboli non hanno alcuna intenzionalità al di fuori di una mente, esse non sono nemmeno manipolazioni di simboli, dato che i simboli, avulsi da un significato, non rappresentano nulla.

Searle, dunque, si inserisce nel contesto di quella che abbiamo precedentemente definito IA Debole.

Paul M. e Patricia Smith Churchland (1990), ribattendo a Searle, sostengono che, essendo il cervello una macchina connessionista, non è assolutamente necessario che al suo interno siano rappresentati simboli, né che vi siano strutture atte a trattarle, poiché è dal parallelismo dei processi che emergono le immagini della mente.

### **3.1.4 - Il test della stanza Cinese**

*I fondamenti teorici di molta parte dell'IA Forte sono la teoria rappresentazionale della mente e l'equivalenza formale tra cervello e computer intelligente. Queste due teorie hanno avuto una larghissima diffusione nell'ambito di studio della filosofia della mente per oltre quarant'anni e sono tuttora sostenute, in varie forme, da molti studiosi che operano nel campo dell'IA.*

*Tra loro uno dei più autorevoli è il filosofo John Searle, che ha sempre confutato la teoria secondo cui il pensiero umano sia riconducibile ad una semplice elaborazione di simboli. La sua argomentazione propone un test di intelligenza appena differente da quello di Turing: il test della stanza cinese.*

*Immaginiamo di prendere una persona qualsiasi che non sappia una parola di cinese. La persona in questione viene chiusa in una stanza in cui ci sono dei fogli di carta con tutti gli ideogrammi cinesi ed un manuale che insegna a combinarli, senza spiegarne il significato.*

*Ora si immagini che fuori della stanza ci sia un cinese che pone delle domande attraverso la porta per mezzo degli stessi fogli con ideogrammi. Se le regole del manuale fossero così precise e dettagliate da permettere all'interrogato di combinare in modo corretto i simboli, l'interrogante avrebbe l'impressione di parlare con un cinese, ma l'interrogato non avrebbe la minima idea del significato delle sue risposte.*

*La metafora di Searle vuole dimostrare che anche se fosse possibile trovare tutte le regole che ci permettono di parlare una lingua, o di ragionare, o di esercitare ogni altra attività intelligente, il computer non sarà mai in grado di capire quello che sta facendo, perché, secondo Searle il computer manipola simboli esclusivamente in base alla loro forma o sintassi e non in base al loro significato.*

*Il significato, infatti, è una proprietà primaria che deriva ai simboli dal fatto di essere usati con riferimento al mondo esterno e questa facoltà di riferimento al mondo è possibile solo per un cervello che stia dentro un corpo in grado di entrare in relazione con l'ambiente che lo circonda. Dal paradosso della Stanza Cinese, Searle estrapola tre assiomi:*

- 1. I programmi di un calcolatore sono formali (sintattici).*
- 2. La mente umana ha contenuto semantico.*
- 3. La sintassi non è condizione né necessaria, né sufficiente per la determinazione di una semantica.*

*La conclusione che deriva da queste tre affermazioni e che riassume il pensiero di Searle, in ambito di IA, è che un calcolatore opportunamente programmato non è in grado di sostituire una mente, perché un programma non è condizione né sufficiente, né necessaria affinché sia data una mente.*

*La teoria di Searle è, dunque, in netto contrasto con le teorie connessioniste classiche, secondo le quali le rappresentazioni mentali non corrispondono in alcun modo a simboli preesistenti in qualche area cerebrale, bensì sono il frutto di elaborazioni che avvengono all'interno del cervello tramite i processi fisici e chimici che si svolgono al livello delle connessioni neuronali.*

*L'interpretazione connessionista della mente sembra, inoltre, giustificare l'aspetto soggettivo della percezione della realtà (Parisi 2000), quello che spiega il possibile divario fra le rappresentazioni mentali che ciascuno può trarre, ad esempio, dalla vista dello stesso oggetto.*

*Se la mente contenesse tutto lo scibile sotto forma di rappresentazione simbolica, questo fenomeno non avrebbe alcuna ragione di esistere, perché ogni manifestazione dell'ambiente esterno corrisponderebbe unicamente ad un simbolo mentale e la realtà fisica assumerebbe un'unica veste indipendentemente da chi se ne facesse interprete.*

*Si porrebbe, poi, il problema di come il cervello potrebbe contenere tutte le informazioni e, procedendo oltre, come si giustificerebbero le diverse capacità di apprendimento, le differenti attitudini ed "intelligenze" degli individui?*

### **3.1.4 – Il ragionamento e la soluzione dei problemi.**

Nell'ambito dell'IA si è tentato di riprodurre sul computer alcune delle facoltà simboliche che caratterizzano un comportamento intelligente, come l'uso del linguaggio naturale e l'organizzazione della conoscenza mediante strutture rappresentazionali.

Questi due concetti, però, non esauriscono assolutamente il significato che ciascuno di noi attribuisce al concetto di intelligenza. Infatti, un essere intelligente deve anche essere in grado di elaborare conoscenze ulteriori a quelle di cui è

dotato, di orientare il suo comportamento in base alle diverse necessità e di risolvere i problemi che gli si presentano.

L'analisi del modello soggiacente al comportamento di un individuo che debba prendere una decisione e la sua riproduzione mediante l'utilizzo di un computer, è alla base di molte ricerche che si collocano sotto la definizione di *problem solving*.

La soluzione dei problemi si applica a contesti molto specifici, ma è possibile individuare un corpo comune di metodologie, che consistono nella ricerca di procedimenti generali per la risoluzione di problemi formalmente definibili.

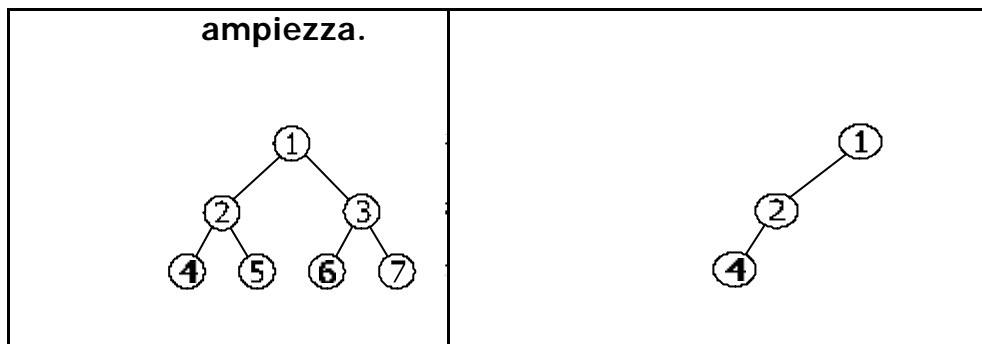
Il primo passo verso la soluzione di un problema è la sua definizione attraverso la determinazione di uno *spazio del problema*. La ricerca di una soluzione è costituita dall'esplorazione dell'intero spazio. Tale procedimento può facilmente essere rappresentato mediante un grafo ad albero, costituito da un insieme di nodi, connessi da segmenti orientati e tale che sia individuabile un nodo di inizio e più nodi terminali, che rappresentano le possibili soluzioni del problema.

La ricerca di una soluzione corrisponde all'esplorazione di ogni ramo dell'albero, finché non si raggiunge un nodo terminale che risponde alle caratteristiche dell'obiettivo della ricerca. Questo percorso si svolge secondo due modalità, che sono la *ricerca in ampiezza* e la *ricerca in profondità*.

Il percorso in profondità prevede che ogni ramo dell'albero venga esplorato sistematicamente dalla radice ad una foglia, se non si giunge ad una soluzione si torna indietro e si esplora un altro ramo, la ricerca si interrompe al raggiungimento del primo obiettivo soddisfacente.

Il percorso in ampiezza, invece, prevede che ad ogni passo siano analizzati tutti i nodi raggiungibili da quello di partenza e ciò avviene per ogni livello del grafo; in questo caso la soluzione accettata sarà la migliore fra tutte quelle possibili, ma può accadere che per raggiungere tale risultato sia stato necessario un enorme dispendio di energie, al contrario il percorso in profondità permette una ricerca più efficiente in termini temporali, ma non garantisce una pari qualità della soluzione.

<b>Ricerca in</b>	<b>Ricerca in profondità.</b>
-------------------	-------------------------------



**Fig. 3.1** - Grafo ad albero con indicazione dei percorsi in ampiezza ed in profondità.

Dati gli inconvenienti che sia la ricerca in ampiezza sia quella in profondità possono presentare, in termini di efficienza ed efficacia, sono successivamente state elaborate delle strategie di ricerca selettiva, definite *euristiche*.

Fra i procedimenti euristici si segnala l'analisi mezzi-fini elaborata da A.Newell, C.Shaw e H.Simon durante l'elaborazione di un programma per la risoluzione generale dei problemi (GPS).

Questo tipo di ricerca procede tentando di ridurre la differenza fra l'obiettivo finale e lo stato iniziale del problema, tramite la formulazione di obiettivi parziali successivi e l'applicazione di opportune regole di trasformazione.

In realtà questo procedimento si è rivelato troppo semplicistico ed ha ricevuto critiche in due direzioni (Haugeland 1988), in primo luogo perché con il GPS non si considerava che ogni classe di problemi necessita di euristiche particolari che non è possibile generalizzare ed in secondo luogo perché si pensava di poter prescindere dalla formalizzazione dei problemi.

### **3.1.5 – I sistemi esperti.**

All'inizio degli anni settanta le ricerche nell'ambito del *problem solving* e, in particolare, la restrizione del campo di competenza per la ricerca delle soluzioni diedero lo stimolo alla creazione dei sistemi esperti, un'importante applicazione pratica dell'intelligenza artificiale.

Semplificando, con il termine "sistema esperto" si indica un programma che è in grado di risolvere problemi complessi in un campo di competenza molto ristretto, con un'efficienza paragonabile a quella di uno specialista umano di quello specifico settore.

Gli elementi fondamentali di ogni sistema esperto sono una base di conoscenza specialistica, un motore inferenziale ed un'interfaccia utente.

La base di conoscenza, per garantire l'efficacia del programma, deve essere molto approfondita ma limitata ad un determinato campo e, soprattutto, rappresentata nella memoria del calcolatore secondo i formalismi che il calcolatore stesso è in grado di interpretare.

Il motore inferenziale, che rappresenta il nucleo del programma, partendo dalla base di conoscenza, è in grado di giungere alla soluzione di un dato problema mediante l'applicazione di procedure euristiche specifiche per l'ambito di applicazione a cui si riferisce il programma.

Altrettanto importante, dal punto di vista dell'efficienza, è l'interfaccia per l'utente, perché permette ad un essere umano di interagire con il programma interrogandolo e leggendo le risposte.

I sistemi esperti hanno avuto una diffusione rilevante soprattutto in campo industriale ed in campo medico, uno dei primi in uso, Mycin, fu introdotto alla Medical School di Stanford per la diagnosi delle malattie infettive nel 1972.

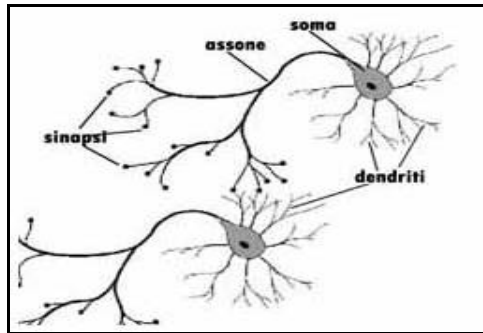
Il programma, tuttora in uso in alcuni ospedali statunitensi, si serve di una base di conoscenza molto dettagliata sulla sintomatologia di tutte le sindromi infettive conosciute e le confronta con la cartella clinica e con l'anamnesi del paziente sotto esame, dopo aver effettuato il confronto, Mycin, elabora la diagnosi e, grazie al procedimento euristico di cui è dotato, fornisce la valutazione del grado di attendibilità delle possibili diagnosi proposte.

Benché i sistemi esperti riescano a raggiungere abilità elevatissime nella propria sfera di competenza, rimangono notevolmente limitati per via della specificità del campo di applicazione, inoltre, mancano di uno degli aspetti fondamentali di un'intelligenza, sia essa naturale o artificiale, cioè della capacità di apprendere ed aumentare automaticamente le proprie conoscenze.

### **3.1.6 – Le reti neurali artificiali (RNA).**

**Il cervello è costituito da cellule, chiamate neuroni, dotati di una forma particolare rispetto alle altre cellule del corpo. I neuroni sono costituiti da un corpo centrale, il soma, dal quale si dipartono numerosi filamenti, i dendriti. Uno solo dei dendriti è più lungo degli altri, l'assone; esso termina in una o più ramificazioni alla cui estremità si trovano le sinapsi, attraverso le quali ciascun neurone è collegato a molti altri neuroni. I neuroni del cervello, quindi, formano un fittissimo reticolo di interconnessioni, attraverso le quali si scambiano vicendevolmente milioni di informazioni al secondo. In base a questo modello biologico viene sviluppato il concetto di rete neurale artificiale (RNA).**





**Fig.3.2** – Struttura neuronale.

“Una rete neurale è una struttura formata da un certo numero di unità collegate tra loro da connessioni. Attraverso le connessioni un'unità influenza fisicamente le altre unità con cui è collegata. Le unità hanno alcune caratteristiche essenziali delle cellule nervose, i neuroni del sistema nervoso reale, mentre le connessioni hanno alcune delle caratteristiche essenziali dei collegamenti sinaptici tra neuroni.” Parisi (1999).

Una funzione a rete neurale dal punto di vista formale è una funzione del tipo:

$$Y=f\{B[f(Ax)]\}$$

in cui i due simboli di funzione rappresentano due trasformazioni non lineari,  $y$  rappresenta il vettore di output,  $x$  rappresenta il vettore degli input ed  $A$  e  $B$  le matrici dei pesi.

In particolare è assolutamente indispensabile che il simbolo di funzione più interno sia una funzione non lineare di tipo logistico:

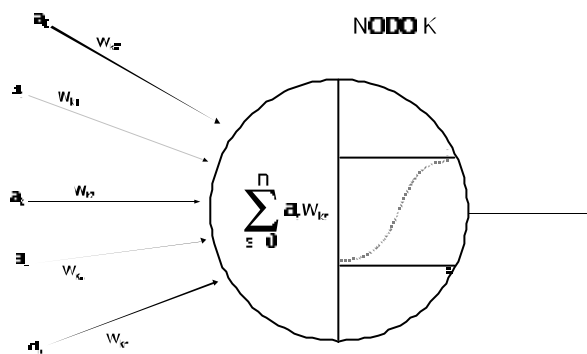
$$f(x)=\frac{1}{1+e^{-x}}$$

La funzione logistica viene scelta in quanto costituisce l'approssimazione di una funzione a soglia, nell'ambito degli studi sul cervello, infatti, si è potuto stabilire abbastanza chiaramente che i collegamenti sinaptici sono di questo tipo, occorre

cioè che gli impulsi elettrici provenienti da un neurone superino una certa soglia per attivare il neurone ad esso collegato.

In questo modo le informazioni, che arrivano direttamente dall'esterno o da altri neuroni, possono viaggiare all'interno del cervello per essere elaborate. Il cammino delle informazioni all'interno di una rete neurale è analogo: un nodo riceve degli input da un nodo precedente o dall'esterno, trasforma l'input ricevuto con una funzione non lineare, la logistica, e restituisce un output che corrisponde all'ordinata della funzione non lineare applicata all'input.

**Per quanto riguarda il generico neurone k, la rappresentazione semplificata della sua struttura interna è la seguente:**



**Fig.3.3** – Struttura di un generico nodo k.

In riferimento all'immagine, i coefficienti  $a_n w_{kn}$  rappresentano le attivazioni che arrivano dall'esterno ponderate con i pesi attribuiti a ciascun legame, infatti, gli indici si riferiscono rispettivamente al nodo di provenienza (n) ed al nodo di destinazione (k).

Ciascun nodo, calcolata la somma degli input, la trasforma con la funzione logistica e restituisce un output pari a zero per un valore della somma inferiore alla soglia e pari a 1 per un valore superiore.

Una rete neurale è costituita da più nodi organizzati su strati diversi, in una RNA abbastanza semplice sono presenti tre strati: uno strato di *input*, uno strato *hidden* ed uno di *output*.

Fra i nodi sono presenti dei legami caratterizzati da pesi, che si autodeterminano durante i cicli di apprendimento, attraverso il metodo della

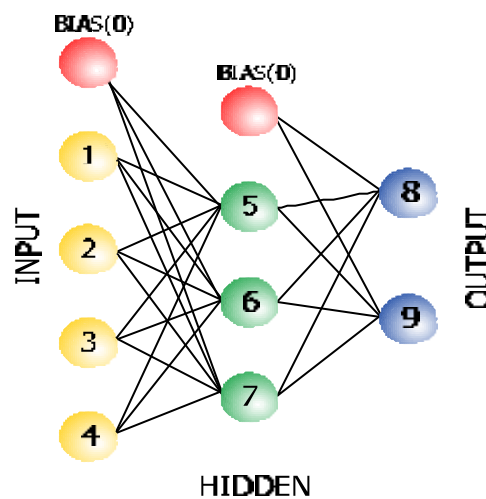
*backpropagation* dell'errore, e possono raggiungere valori tali per cui la RNA è in grado di approssimare un risultato con un errore praticamente nullo.

Nello strato di input ed in quello di *hidden* è possibile inserire dei nodi cosiddetti *bias*, che servono per aumentare i gradi di libertà del sistema. La funzione a rete neurale, in caso di presenza dei *bias*, diventa :

$$Y=f\{B[1,f(A(1,x))]\}$$

I nodi *hidden* sono fondamentali per il funzionamento e l'efficacia dei risultati ottenibili da una RNA, poiché è proprio al livello *hidden* che avviene la trasformazione non lineare, il passaggio, cioè, che distingue una rete neurale da un sistema di equazioni lineari.

Dal punto di vista grafico una rete neurale a tre strati, con *bias* nello strato di input e di *hidden* si presenta come nel grafico seguente:



**Fig.3.4** – Struttura di una rete neurale a tre strati, con 4 nodi di input, 3 nodi hidden, 2 output e 2 nodi bias.

Tornando alla metafora cerebrale, data la sua struttura ed il suo funzionamento, una RNA è in grado di svolgere calcoli paralleli ed interattivi, ed è in grado, inoltre, di conservare in modo distribuito le informazioni, pur essendo molto semplificata, dunque, possiede quelle che abbiamo definito come le caratteristiche basilari di un sistema intelligente.

### 3.2 – Dall'Intelligenza Artificiale all'Alife.

La Vita Artificiale (*Alife*) nasce, come campo di studi riconosciuto dalla comunità scientifica, alla fine degli anni ottanta in occasione di una conferenza organizzata dal biologo Christopher Langton a Los Alamos, in New Mexico.

Questa nuova disciplina nasce dall'incontro fra l'IA classica e le nuove teorie evoluzionistiche e si propone di riprodurre la vita animale in forma artificiale, per mezzo delle metodologie dell'IA classica rivisitate in senso evoluzionistico, come le reti neurali auto-evolventesi.

### **3.2.1 – L'evoluzionismo**

Il concetto di evoluzione a cui l'*Alife* fa riferimento trova i suoi fondamenti nelle teorie di Darwin e Lamarck, nonostante le differenti posizioni concettuali, entrambi hanno contribuito moltissimo nello sviluppo della genetica computazionale.

A Darwin si deve, infatti, il concetto di selezione naturale, secondo cui la Natura fa prevalere gli individui di ogni specie che meglio si adattano all'ambiente circostante e che, quindi, sono maggiormente in grado di sopravvivere e di riprodursi.

Il concetto di selezione naturale prevede che in ogni momento ed in ogni specie possano presentarsi individui con caratteristiche proprie ed emergenti, che producono diversità, e che si adattano all'ambiente meglio o peggio rispetto agli individui precedenti. Quindi le mutazioni favorevoli alla sopravvivenza tendono ad affermarsi, mentre quelle sfavorevoli conducono all'estinzione della specie.

Secondo Lamarck, invece, "la funzione crea l'organo", alcuni individui di una specie sviluppano migliori capacità di adattamento all'ambiente e le tramandano nella riproduzione ai propri successori, in questo modo in ogni specie si affermano progressivamente solo le caratteristiche migliori per la sopravvivenza.

### **3.2.2 – Algoritmi genetici.**

Gli algoritmi genetici, formalizzati da Holland (1975), hanno ricevuto una notevole attenzione da parte degli studiosi di scienze sociali e di economia, rivelandosi una valida alternativa nella trattazione di problemi difficilmente risolvibili con i metodi analitici tradizionali.

Holland, partendo da uno dei problemi fondamentali della matematica, l'ottimizzazione delle funzioni, individuò una strada di soluzione efficiente nell'evoluzione genetica e nella selezione naturale. La particolare natura degli algoritmi genetici sta quindi nel concetto della sopravvivenza del migliore, o meglio, del più adatto alla sopravvivenza nell'ambiente in cui vive.

In Goldberg (1989), "Gli algoritmi genetici sono algoritmi di ricerca basati sui meccanismi di selezione naturale e sulla genetica. Integrano un principio di sopravvivenza della struttura maggiormente adatta all'ambiente con un meccanismo di scambio di informazioni, strutturato ma stocastico, per formare un algoritmo di ricerca che presenta un po' dell'intuito innovativo insito nella ricerca umana. Ad ogni generazione, un nuovo insieme di creature artificiali (sequenze) è generato sulla base degli individui migliori dell'insieme precedente (...). Seppur stocastici, gli algoritmi genetici non effettuano una semplice ricerca casuale. Sfruttano invece in modo efficiente le informazioni storiche per orientarsi verso nuovi punti di ricerca caratterizzati da una migliore prestazione attesa."

Gli algoritmi genetici sono algoritmi di ottimizzazione ed apprendimento automatico, basati sul completamento artificiale di alcuni principi fondamentali della selezione naturale e della genetica evolutiva.

I parametri del problema da affrontare vengono codificati in stringhe di informazioni a cui viene associato un valore di adattività o *fitness*, che vengono poi fatte riprodurre mediante copia, mutazione o incrocio. Il primo passo per la progettazione di un AG è la determinazione dello schema binario con il quale rappresentare tutte le informazioni necessarie per la definizione del problema.

In seguito viene generata una popolazione in grado di esplorare parallelamente lo spazio delle soluzioni e di cui si valutano le prestazioni individuali tramite l'attribuzione di un valore di *fitness*. Infine, in base al valore di fitness di ciascun individuo, gli si attribuisce una probabilità di riproduzione.

Il meccanismo di base degli AG è sintetizzato efficacemente in Koza (1990): "Gli algoritmi genetici sono algoritmi matematici ad alto parallelismo che trasformano popolazioni di oggetti matematici individuali (in genere sequenze binarie di lunghezza fissa) in nuove popolazioni utilizzando operatori mutuati dalle operazioni genetiche naturali come la riproduzione sessuata (*cross-over*) e dalla riproduzione proporzionale al grado di adattamento (principio di adattamento del migliore di Darwin).

Gli algoritmi genetici partono da una popolazione iniziale di individui (generata casualmente) e iterativamente valutano il grado di adattamento all'ambiente di ogni individuo della popolazione e applicano operatori genetici su diversi individui della popolazione per generare una nuova popolazione".

Analizzando nel dettaglio questo meccanismo occorre fare alcune precisazioni riguardo agli elementi fondamentali, la generazione della popolazione iniziale, la valutazione degli individui, la riproduzione e l'applicazione degli operatori genetici.

La popolazione iniziale viene generata solitamente in modo casuale, sulla base della rappresentazione del problema ed è costituita da un numero variabile di individui indipendenti. La numerosità della popolazione può essere costante durante tutta la simulazione oppure può variare all'atto delle riproduzioni e del passaggio da una generazione di individui a quella successiva.

In base al patrimonio genetico degli individui e del loro adattamento all'ambiente, in base alla codifica del problema da risolvere, a ciascun individuo viene attribuito un valore di *fitness*.

Questo valore è di fondamentale importanza al momento della selezione degli individui per la riproduzione, poiché in un'ottica di selezione Darwiniana gli individui migliori hanno più probabilità di riprodursi.

E' altrettanto importante, però, per evitare di raggiungere delle soluzioni svianti, che non compaiano subito individui con una fitness troppo elevata, i quali conducano la popolazione ad evolversi in direzione di soluzioni non ottimali.

La modalità di selezione che viene scelta e la tecnica di riproduzione rappresentano due aspetti rilevanti per l'evoluzione dell'algoritmo verso una soluzione del problema a cui è applicato il suo funzionamento.

Il principale obiettivo della selezione è quello di attribuire maggiori probabilità di riproduzione agli individui che presentano un grado di adattamento più elevato.

Per questo motivo uno dei sistemi di selezione più utilizzati è quello della *wheel selection*, che attribuisce a ciascun individuo una probabilità di riproduzione proporzionale al proprio livello relativo di fitness. Successivamente all'individuazione del criterio di selezione, occorre stabilire se la popolazione viene sostituita ad ogni generazione o se solo alcuni con fitness bassa individui vengono sostituiti.

Il punto cruciale dello sviluppo di un AG si raggiunge, però, solo nell'applicazione degli operatori genetici, infatti senza questo passo si avrebbe semplicemente un'esplorazione dello spazio delle soluzioni in base agli individui con le caratteristiche iniziali. Nel momento in cui vengono introdotti gli operatori genetici, invece, si creano individui con caratteristiche nuove e, spesso, inattese.

I principali operatori genetici utilizzati negli AG sono il *crossover* e la mutazione, benché alcuni studiosi, fra cui Goldberg e lo stesso Holland, considerino solo il *crossover* come veramente efficace.

La tecnica di *crossover* consiste in una sorta di riproduzione sessuata fra individui, che coinvolge due "genitori" che fanno confluire il proprio patrimonio genetico in altrettanti "figli", ognuno dei quali avrà parte del patrimonio genetico dei due individui iniziali.

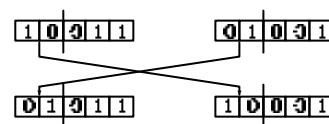


Fig 3.5 - Cross-over

La mutazione, più semplicemente, consiste in un errore di trascrizione di un carattere del genotipo dell'individuo con conseguente modifica delle caratteristiche dell'individuo.

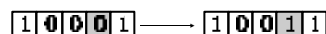


Fig.3.6 - Mutazione

### 3.2.3 – Classifier System.

"Un *classifier system* è un sistema di apprendimento automatico che impara delle regole sintatticamente semplici (chiamate *classifier*) per guidare le sue azioni in un ambiente arbitrario" in Goldberg (1989).

I *classifier system* possono essere considerati dei successori dei sistemi esperti basati su regole, benché si differenzino da essi grazie ad alcune caratteristiche fondamentali, che sono il parallelismo nella generazione delle regole,

l'apprendimento automatico dei crediti da attribuire alle regole in base ad un sistema competitivo e soprattutto la creazione di regole, effettuata autonomamente per mezzo di algoritmi genetici.

Grazie all'applicazione degli Algoritmi genetici i *classifier system* permettono, però, di superare quello che era stato lo scoglio principale nell'uso dei sistemi esperti e che ne aveva impedito l'utilizzo in campi di ricerca particolarmente complessi, cioè la definizione di un enorme numero di regole di funzionamento ed una sintassi di queste regole particolarmente complicata.

### 3.2.3.1 - Struttura e funzionamento.

Un sistema a classificatori è costituito da tre elementi fondamentali, un sistema di regole, un sistema di attribuzione dei crediti ed un algoritmo genetico. Accanto a questi componenti vi sono dei dispositivi di input ed output la cui funzione è, rispettivamente, quella di codificare l'informazione proveniente dall'ambiente in forma di rappresentazioni manipolabili dal *classifier (detector)* e quella di convertire queste rappresentazioni interne di regole in azioni compiute dal *classifier* stesso sull'ambiente (*effector*).

Tale struttura viene detta sistema a classificatori perché al suo interno si evolvono regole che classificano gli input associandovi un opportuno output secondo uno schema del tipo **se** <condizione>, **allora** <azione>.

La forma rappresentazionale utilizzata da un *classifier system* per i messaggi è quella di una stringa binaria, mentre le rappresentazioni dei classificatori sono stringhe in grado di codificare le caratteristiche possibili dell'input, nonché le azioni possibili da intraprendere, secondo un alfabeto ternario composto dai simboli {0,1,#}. In questa codificazione particolare, l'1 indica la presenza di una caratteristica, lo 0 ne indica l'assenza e il simbolo # indica che quella caratteristica può valere indifferentemente 0 oppure 1.

L'attivazione di un classificatore in corrispondenza di un messaggio di input dipende da quanto quest'ultimo sia simile alla stringa <condizione> della regola di produzione.

Al termine del *matching* fra messaggio e regola, viene attivato un meccanismo di scelta della regola che deve agire o proporre la sua azione tramite l'*effector*.



Per poter agire ogni regola deve sostenere un costo, detratto dal suo patrimonio di crediti, una volta comunicata all'ambiente dall'*effector* la regola riceve una valutazione, in base alla quale ottiene un premio, che viene ripartito fra tutte le regole che hanno concorso a creare l'azione.

L'iterazione del procedimento permette alle regole migliori di accumulare crediti, mentre impoverisce le regole meno buone, la successiva applicazione di un algoritmo genetico che seleziona in base alla "ricchezza" di ogni regola permette al sistema di far evolvere ad ogni ciclo solo le regole che si sono rivelate migliori e causa l'estinzione delle peggiori.

In questo modo le popolazioni che si formano sono sempre più in grado di avere interazioni con l'ambiente sono sempre più abili nella soluzione dei problemi a cui sono applicate.

#### **3.2.3.2 - I premi e le penalità, l'algoritmo *bucket brigade*.**

L'algoritmo *bucket brigade* fu ideato dallo stesso Holland ed è basato sul pagamento dei premi e delle penalità alle regole che hanno attivato le azioni sull'ambiente.

Una curiosità riguarda l'origine del nome, che indica il sistema di passaggio dei secchi d'acqua che usavano i pompieri per spegnere gli incendi, secondo la modalità *bucket brigade*, infatti, i secchi venivano passati di mano in mano fino al primo della fila e ritornavano indietro con lo stesso sistema, ma vuoti e pronti per essere nuovamente riempiti.

Le regole che hanno agito direttamente sull'ambiente pagano, a ritroso, una quota dei propri crediti a quelle regole che ne hanno permesso l'attivazione, cioè che hanno prodotto a loro volta un'azione in grado di verificare il *match* con la regola stessa.

Ogni regola si inserisce in una catena di creditori e debitori, ogni volta che ottiene il diritto ad agire, deve pagare una quota del proprio patrimonio ai suoi "fornitori", in questo modo ciascuna regola può aumentare il proprio patrimonio sia perché ha potuto agire sull'ambiente in modo giudicato positivo, sia perché ha permesso l'attivazione di un'altra regola.

### **3.2.4 – Automi cellulari.**

Gli automi cellulari sono costituiti da piccoli mondi artificiali di cellule, la cui evoluzione è autonomamente governata da regole deterministiche.

Essi rappresentano un'ulteriore tappa nel tentativo di replicare artificialmente fenomeni come l'autoriproduzione, che fino a poco tempo fa era considerata dominio esclusivo del regno biologico.

Basandosi sul lavoro di Turing sugli automi finiti e sulla morfogenesi e su quello di Ulam sul comportamento dei sistemi complessi, John Von Neumann, a cavallo fra gli anni '40 e '50 formulò il modello di una macchina autodiretta (un automa) esistente in un piano bidimensionale, che era in grado di riprodursi e di risolvere problemi computabili: un automa cellulare.

I modelli tradizionali di computazione, come la macchina di Turing, mantengono una netta divisione tra la parte strutturale di un computer ed i dati su cui esso opera, il computer non è in grado, quindi, di agire su se stesso e sulla materia di cui è costituito. In un automa cellulare, invece, oggetti attivi e passivi di computazione sono assemblati in un'unica struttura e sono soggetti alle stesse regole di evoluzione del sistema: la computazione e la costruzione sono solo due possibili stati del sistema.

In termini più rigorosi un automa cellulare è un sistema dinamico a stati discreti il cui comportamento è completamente determinato dall'azione, a livello locale delle singole celle che lo compongono, di regole deterministiche stabilite globalmente per il sistema, cioè le regole di evoluzione dell'automa.

Un automa cellulare può essere considerato un mondo in miniatura stilizzato attraverso una simulazione al computer, in cui lo spazio è rappresentato da una griglia uniforme, le cui celle possono essere vuote oppure occupate da cellule dotate di caratteristiche predeterminate.

Il tempo viene rappresentato in stati discreti e le regole di evoluzione del sistema sono espresse da tabelle di stato, mediante le quali ciascuna cellula computa il suo stato successivo in base a quello delle cellule adiacenti.

Le regole del sistema, dunque, agiscono a livello locale, uniforme e parallelo su tutte le celle della griglia, fornendo in questo modo modelli attendibili per molte

ricerche su fenomeni naturali, in particolare per lo studio, attraverso la simulazione artificiale, dell'evoluzione di grandi sistemi fisici o biologici.

### **3.2.5 – *Life e Tierra*, due esempi.**

L'automa cellulare *Life* fu creato dal matematico John Horton Conway intorno alla metà degli anni '60 e può essere considerato la simulazione di un processo biologico di evoluzione cellulare.

Lo spazio computazionale di questo automa è costituito da una griglia toroidale (il bordo superiore dello spazio rappresentato sul piano coincide con quello inferiore ed il bordo destro coincide con quello sinistro) e per ogni cella della griglia sono possibili due stati: la cella può essere viva o morta.

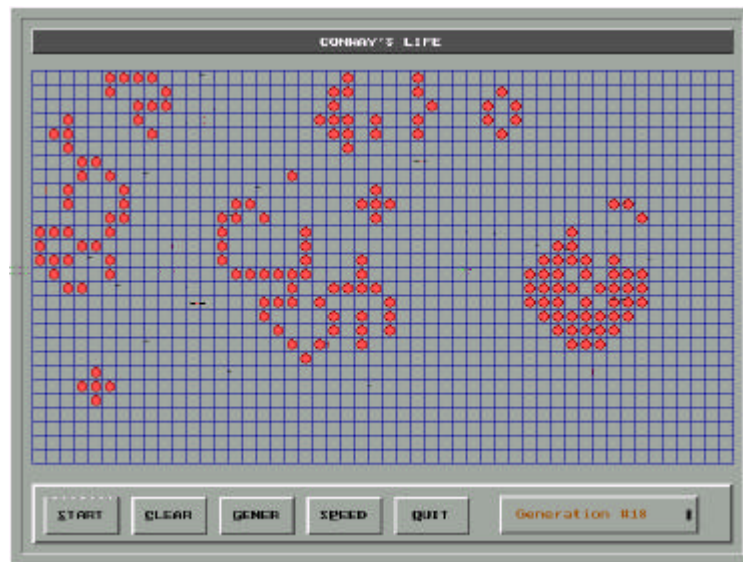
Il funzionamento dell'automa dipende dall'analisi dello stato di ogni cella rispetto al proprio "vicinato", infatti l'intorno di ogni cella è costituito da otto celle adiacenti, quattro ortogonalmente e quattro diagonalmente (il cosiddetto "vicinato di *Moore*") e dal comportamento che segue a questo tipo di analisi.

Le regole che determinano il comportamento dell'automa sono molto semplici, infatti ad ogni generazione muoiono le celle vive con solo una o due celle adiacenti vive (morte per isolamento) e muore anche ogni cella con quattro celle adiacenti vive (morte per affollamento), mentre le celle morte con tre celle vive adiacenti tornano in vita al ciclo successivo.

Dopo un certo numero di cicli il sistema può morire per l'estinzione delle celle vive, può crescere assumendo forme nuove e, forse, inattese, oppure può stabilizzarsi in figure che, per le regole imposte, tenderanno a non modificarsi, o a riprodursi identiche, come gli alianti o i quadrati.

In uno studio sugli automi cellulari, Stephen Wolfram (19xx), stabilì una classificazione degli automi mono e bi-dimensionali, sulla base dell'osservazione del loro comportamento evolutivo e della loro maggiore o minore organizzazione in configurazioni strutturate di cellule, individuando quattro diversi automi.

Wolfram distinse fra automi semplici, che evolvono verso uno stato stabile in cui tutte le celle assumono lo stesso stato; automi stabili, caratterizzati dalla presenza di configurazioni stabili (fisse) o instabili (oscillatorie) di celle; automi disorganizzati, caratterizzati dall'assenza di configurazioni strutturate di celle ed, infine, automi caotici, che mostrano un comportamento complesso, caratterizzato da un susseguirsi di configurazioni strutturate ed assembramenti disorganizzati di celle.



**Fig.3.7** - Automa Life di Conway, in cui si può notare una sola struttura cellulare stabile.

Il programma *Tierra*, creato dal biologo evoluzionista Thomas Ray, simula un mondo popolato da programmi autoreplicanti che devono dividersi il medesimo spazio vitale ed allo stesso tempo evolvere.

Confrontando la robustezza della codifica del codice genetico, con la rigidità dei codici di linguaggio dei computer, Ray stabilì che per consentire la riproduzione dei programmi mediante mutazione e ricombinazione (*crossing-over*) occorre ridurre il set di istruzioni, stabilendo in 5 bit le dimensioni dell'unità di informazione.

Il problema che si pose successivamente era legato alla struttura funzionale dei programmi che, ripetendo le loro routine di funzionamento, scrivono e leggono le informazioni in zone della memoria anche molto distanti fra loro di cui devono specificare le coordinate, per ovviare alla sovrabbondanza di informazioni necessarie, Ray codificò le molecole in modo tale che ciascuna avesse una certa probabilità di venire in contatto con il più vicino modello avente una superficie complementare ad esso.

Dopo un sufficiente numero di cicli, alcune creature artificiali svilupparono tecniche per sfruttare al meglio l'ambiente, mentre le altre mettevano in opera strategie per contrastare le prime.

Dopo molti cicli si delinearono due tipi principali di creature, uno che tratteneva dentro di sé ogni informazione per sopravvivere e rubava anche quella

dell'ambiente, l'altro, che per vivere e riprodursi, copiava ad ogni ciclo tutte le informazioni necessarie.

Col procedere del programma, le creature del primo tipo, non copiando le informazioni necessarie alla riproduzione, si trovarono in difficoltà, infatti, quando non trovarono più abbastanza individui dell'altro tipo a cui carpire le informazioni, iniziarono a morire, mentre quelle dell'altro genere ripresero lentamente a proliferare ed a passare informazioni sbagliate, in modo tale che, dopo alcuni cicli, restarono solo creature dello stesso genere ed iniziarono a cooperare per la sopravvivenza.

Secondo il criterio di ereditarietà scoperto da Lamarck, la tecnica della divulgazione delle informazioni sbagliate risultò valida e, come tale, si riprodusse nel mondo artificiale.

In una sperimentazione, eliminata la possibilità di mutazione venne osservato che le creature continuavano ad evolversi grazie ad una forma di riproduzione sessuata, che permetteva loro di scambiare parti di patrimonio genetico.

L'aspetto fondamentale è che nessuna delle strutture che si erano evolute nel tempo facevano parte dello stadio iniziale di *Tierra*, ma si erano evolute autonomamente, senza alcuna influenza dall'esterno.

### **BIBLIOGRAFIA CAPITOLO TERZO.**

- CHOMSKY N.(1979), *Saggi Linguistici*,vol.2. *La grammatica generativa trasformativale*, Bollati Boringhieri, Torino.
- CHURCHLAND P.M., CHURCHLAND SMITH P. (1990), *Può una macchina pensare?*, California University San Diego.
- HAUGELAND J. (1989), *Progettare la Mente. Filosofia, psicologia, intelligenza artificiale*, Il Mulino, Bologna.
- HAUGELAND J. (1988), *Intelligenza Artificiale*, Bollati Boringhieri, Torino.
- HOBBS T. (1982), *The Leviathan*, (tr. Italiana) *Il Leviatano*, Editori Riuniti, Roma.
- PARISI D. (1999), *Mente. I nuovi modelli della vita artificiale*, Il Mulino, Bologna.
- SEARLE J. (1980), *Menti, cervelli e programmi*, Berkley.
- TERNA P. (1995), *Paper XXXV Riunione scientifica annuale della Società Italiana degli Economisti*.
- TURING A. (1994), *Macchine Calcolatrici e Intelligenza*, IN V. SOMENZI, R. CORDESCHI, *La Filosofia degli Automi*, Bollati Boringhieri, Torino.
- **Weizenbaum J. (1987), *Il potere del computer e la ragione umana*, Gruppo Abele, Torino.**

## Capitolo Quarto: Modelli di simulazione basati su Agenti.

Un modello è una riproduzione della realtà, che può essere fatta in diverse forme, in ambito economico-sociale vengono utilizzati soprattutto modelli matematico-statistici e, solo in epoca recente, si è introdotto l'uso dei modelli simulativi, per tentare di riprodurre quelle situazioni che, per via della loro natura, possono difficilmente essere trattate con i metodi computazionali tradizionali.

In Axtell (2000), *"The many motivations for employing agent-based computation in the social sciences are reviewed. It is argued that there exist three distinct uses of agent modeling techniques. One such use — the simplest — is conceptually quite close to traditional simulation in operations research. This use arises when equations can be formulated that completely describe a social process, and these equations are explicitly soluble, either analytically or numerically. In the former case, the agent model is merely a tool for presenting results, while in the latter it is a novel kind of Monte Carlo analysis. A second, more commonplace usage of computational agent models arises when mathematical models can be written down but not completely solved. In this case the agent-based model can shed significant light on the solution structure, illustrate dynamical properties of the model, serve to test the dependence of results*

*on parameters and assumptions, and be a source of counter-examples. Finally, there are important classes of problems for which writing down equations is not a useful activity. In such circumstances, resort to agent-based computational models may be the only way available to explore such processes systematically, and constitute a third distinct usage of such models."*

L'uso dei modelli ad agenti permette di ottenere alcuni vantaggi rispetto ai modelli delle teorie matematiche convenzionali, in primo luogo semplifica l'utilizzo di agenti a razionalità limitata, permettendo di superare i concetti classici come ad esempio "aspettative razionali", "effetti di sostituzione", "avversione al rischio" e "preferenze", in secondo luogo, anche se si sceglie di utilizzare agenti razionali, diventa semplice creare agenti eterogenei, semplicemente differenziando la popolazione in base a diverse distribuzioni degli stati iniziali, ad esempio il tipo di preferenze, se il modello si riferisce ad un mercato di consumatori.

In terzo luogo, dato che il modello raggiunge una soluzione durante l'esecuzione, il risultato può essere costituito dall'intera dinamica di svolgimento del processo studiato, consentendo di focalizzare la propria attenzione, non solo sullo stato di equilibrio eventualmente esistente, ma soprattutto sulle dinamiche che ne consentono il raggiungimento.

Infine, i modelli ad agenti permettono, grazie ad una particolare versatilità ed alla scarsa necessità di formalismi matematici evoluti, la rappresentazione anche di processi sociali difficilmente rapportabili alla matematica tradizionale.

In ogni caso i modelli ad agenti hanno uno svantaggio rilevante rispetto ai modelli matematici convenzionali, strettamente legato con la loro natura dinamica, infatti, sono necessari numerosi cicli prima di dimostrare la convergenza dei comportamenti individuali alla rappresentazione statica e formale di un teorema, perché un unico ciclo non è reputabile esaustivo, a differenza di un modello rappresentato, ad esempio, da un sistema di equazioni.

In Axtell (2000) *"It has been argued that there exist three distinct uses of agent-based computational models. First, when numerical realizations are relevant, agents can perform a variant of classical simulation. Second, when a model is only incompletely solved mathematically – its equilibria unknown, stability of equilibria undetermined, or the dependence on parameters opaque – then an agent-based model can be a useful tool of analysis, a complement of mathematics. Third, there are cases in which mathematical models are either apparently intractable or provably insoluble. In such circumstances it would seem that agent computing is perhaps the only technique available for systematic analysis, a substitute for formal mathematics analysis."*

Molti dei problemi considerati dalle scienze sociali, probabilmente, ricadono proprio in quest'ultima definizione, essendo difficilmente riconducibili ad una trattazione matematica pura ed esaustiva, poiché la modellizzazione matematica dei processi economici non è una via perseguibile, senza l'uso di numerose semplificazioni che limitano il realismo e la plausibilità della rappresentazione stessa.

Se le tecniche tradizionali di analisi matematica non rappresentano un metodo sufficientemente versatile per lo studio delle scienze sociali, i modelli ad agenti pongono un'altra



serie di limiti determinati dagli strumenti utilizzati e dalla necessità di creare uno standard, in Rust (1996) " The reason why large computable general equilibrium problems are difficult for economists to solve is that they are using the wrong hardware and software. Economists should design their computations to mimic the real economy, using massively parallel computers and decentralized algorithms that allow competitive equilibria to arise as «emergent computations»... The most promising way for economists to avoid the computational burdens associated with solving realistic large scale general equilibrium models is to adopt an "agent-based" modeling strategy where equilibrium prices and quantities emerge endogeneously from the decentralized interactions of agents."

Prendendo a modello le formiche ed altri insetti sociali, gli informatici hanno creato agenti software che cooperano per risolvere problemi complessi, gli insetti che vivono in colonie, infatti, sembrano agire in modo totalmente autonomo, eppure il gruppo, nel suo insieme, appare altamente organizzato.

In apparenza, l'integrazione di tutte le attività individuali non richiede alcuna supervisione, in realtà, lo studio del comportamento degli insetti sociali ha rilevato che la cooperazione a livello di colonia è in gran parte auto-organizzata, cioè deriva dalle interazioni fra gli individui.

Anche se queste interazioni possono essere semplice, ad esempio una formica segue semplicemente una traccia di feromone lasciata da un'altra, nell'insieme permettono di risolvere problemi complessi, infatti la stessa formica può individuare la strada più breve che la separa dal cibo fra innumerevoli itinerari possibili.

Questo comportamento collettivo, che emerge da un gruppo di insetti sociali, è stato definito *swarm intelligence* o intelligenza collettiva, in Bonabeau, Dorigo e Theraulaz (1999), ed ha suggerito metodi nuovi per lo studio di fenomeni collettivi in campi anche molto diversi fra loro, dall'instradamento delle chiamate in sistemi di telecomunicazioni ad alto traffico, alla robotica, ai sistemi di suddivisione del lavoro nelle catene di montaggio.

L'intelligenza collettiva focalizza la sua attenzione soprattutto sull'autonomia e l'autosufficienza dei sistemi, basandosi su relazioni dirette ed indirette fra agenti semplici, per questo la riproduzione di ambienti artificiali che simulano quelli reali, necessita di sistemi adatti a ricreare questo tipo di interazione, non coordinata da un'entità dotata di controllo sull'insieme, ma auto-organizzata e generata dall'agire simultaneo e ripetuto di individui che non hanno visione dell'insieme in cui si muovono, ma soltanto delle proprie azioni e di quelle degli individui con cui vengono in contatto.

Il progetto Swarm (1994) nasce esattamente con questo intento, in Johnson, Lancaster e Stefansson (1999) "The Swarm project was started in 1994 by Chris Langton at Santa Fe Institute (SFI) in New Mexico. The aim was to develop both a vocabulary and set of standard computer tools for the development of multi-agent simulation models (Agent Based Models). Armed with this framework, researchers are able to focus on the substance of the modeling task, avoiding some of the complicated (and uninteresting) details of computer coding.

#### 4.1.1 – Swarm e la sperimentazione nelle scienze sociali.

*Swarm* è un prodotto software, costituito da un insieme di librerie, programmi e protocolli, appositamente studiato per la gestione di modelli di simulazione comportamentali basati su agenti adattivi. *Swarm*, infatti, è costituito da sciame di agenti, che possono a loro volta contenere altri sciame, e che vengono fatti interagire in base a regole individuali.

Lo *swarm* deve essere considerato come un gruppo strutturato di individui che sono in grado di interagire, a prescindere dalla propria natura, infatti, uno sciame può essere costituito tanto da insetti, nella simulazione di un sistema animale, tanto da operatori economici, per una simulazione sociale.

In Minar, Burkhart, Langton, Askenazi (1996), *"Swarm is a multi-agent software platform for the simulation of complex adaptive systems. In the Swarm system the basic unit of simulation is the "swarm", a collection of agents executing a schedule of actions. Swarm support hierarchical modelling approaches whereby agents can be composed of swarms of other agents in nested structures. Swarm provides object oriented libraries of reusable components for building and analyzing, displaying and controlling experiments on those models"*.

Nell'ambito delle scienze sociali, l'evolversi delle metodologie di studio verso l'uso di modelli comportamentali, il cui principale oggetto di indagine è costituito dalle risposte individuali agli stimoli ambientali, e dai meccanismi attraverso i quali l'interazione dei singoli sfocia in fenomeni complessi, conduce all'impiego di metodi sperimentali veri e propri, che vanno oltre la semplice formalizzazione in modelli matematici e riproducono in una sorta di laboratorio i fenomeni economici e lo sviluppo dei mondi artificiali, reso possibile da *Swarm*, deve essere considerato una strada verso questo orientamento sperimentale.

In un modello *Swarm*, le strutture sociali ed i comportamenti collettivi del gruppo, infatti, emergono dall'interazione degli individui, rappresentati da programmi, che operano nell'ambiente artificiale e compiono azioni basate su regole individuali autonomamente sviluppate o, in parte, imposte dal programmatore.

Questo tipo di impostazione, a differenza delle tecniche di analisi tradizionali, permette di analizzare la formazione di istituzioni e strutture economiche nel momento del loro divenire, cioè quando si manifestano i meccanismi individuali sufficienti a generare i comportamenti collettivi all'origine dei fenomeni economici di interesse dello sperimentatore.

In ambito strettamente economico queste metodologie sperimentali vengono indicate come *Agent based computational economics (ACE)*, di cui si ha una definizione in Testfatsion (1997): *Agent-Based Computational Economics (ACE) is roughly characterized as the computational study of economies modelled as evolve decentralized systems of autonomous interacting agents. A central concern of ACE researchers is to understand the apparently spontaneous appearance of global regularities in economic processes, such as the unplanned coordination of trade in decentralized market economies that economists associate with Adam Smith's "invisible hand". The challenge is to explain these global regularities from the bottom-up, in the sense*

*that the regularities arise from the local interactions of autonomous agents channeled through actual or potential economic institutions rather than through fictitious top-down coordinating mechanism such as a single representative consumer.*

La ACE, dunque, analizza il sorgere dei fenomeni da una dimensione individuale ad una collettiva e complessa e si interroga su quale sia il processo dall'una all'altra, negando le metodologie riduzioniste che analizzavano, invece, il risultato di tali fenomeni e lo riconducevano ad un singolo individuo, considerandolo rappresentativo di un insieme di riferimento.

Secondo la nuova impostazione, quindi, il mercato è un insieme complesso in continua evoluzione e va analizzato in questa prospettiva, rinunciando alla spiegazione delle proprietà del sistema mediante l'analisi delle caratteristiche dei singoli elementi che lo compongono e valutando, invece, l'importanza della dinamica del sistema stesso e delle leggi che ne permettono l'evoluzione nel tempo.

In un modello *Swarm*, ciascun agente possiede una marcata autonomia, poiché è in grado di prendere decisioni sulle azioni da compiere e di adattare il proprio comportamento in base alla percezione che ha dell'ambiente, questo significa che l'agente non è l'esempio rappresentativo di una classe, bensì è un individuo e, come tale, uguale probabilmente solo a se stesso.

L'accento posto sull'individuo conduce alla definizione di "individualismo metodologico" in Axtell ed Epstein (1996), specialmente in riferimento all'assenza, in questo tipo di modelli, di istituzioni predefinite in grado di orientarne l'evoluzione. Pur tuttavia, fra individualismo metodologico e modelli ad agenti non può esservi perfetta coincidenza, infatti, questi ultimi si prestano particolarmente allo studio della nascita di comportamenti emergenti, cioè non previsti e non prevedibili, come risultato dell'interazione fra le varie componenti del sistema.

Il problema che si pone, dunque, riguarda essenzialmente la complessità, e da dove essa scaturisca, lo sperimentatore, all'atto dell'impostazione del modello, deve stabilire se propendere per un'impostazione maggiormente legata al riduzionismo dell'individualismo metodologico e, quindi, includere la complessità negli agenti stessi oppure se optare per l'estrema semplicità degli agenti e lasciare che semplicemente dalla loro interazione emergano comportamenti complessi.

Entrambi i punti di vista comportano degli inconvenienti, infatti, seguendo la prima via si rischia di legare il modello a condizioni iniziali arbitrariamente definite e di incorrere in una relazione di autoreferenza, mentre con la seconda via, probabilmente, non si riesce a far convergere il modello ad alcuna soluzione plausibilmente fondata, ma si genera semplicemente del "rumore bianco".

La via che sembra garantire i migliori risultati è quella intermedia, procedere con agenti ragionevolmente semplici, ricercando nell'interazione e nella presenza di strutture ex-ante i mezzi adatti a riprodurre la reale complessità, che potrà essere interpretata secondo la scala migliore, non necessariamente coincidente con quella individuale.

I modelli ad agenti, la cui creazione viene particolarmente agevolata dall'uso di *Swarm*, sono, quindi, particolarmente adatti a riprodurre situazioni complesse, poiché permettono di scomporre la realtà rappresentata in elementi, che non devono essere necessariamente complicati e neanche devono tentare di riassumere al proprio interno la complessità dell'intero sistema, ma che semplicemente forniscono un contributo alla rappresentazione della realtà ed allo svolgimento del modello stesso.

#### **4.1.2 – Struttura e funzionamento di un modello in Swarm.**

I modelli simulativi condotti mediante la sperimentazione con il computer, sono generalmente costituiti da un ambiente artificiale, popolato da un insieme di oggetti che interagiscono in base a regole elementari coordinate secondo lo spazio ed il tempo.

La maggior differenza fra la realtà rappresentata ed il modello rappresentante è costituita, infatti, proprio dal tempo: se, infatti, nella realtà il tempo scorre naturalmente, senza bisogno di alcun intervento esterno, in un modello di simulazione al computer, invece, il tempo deve essere fatto scorrere artificiosamente, per permettere agli individui di agire in maniera sincronizzata e secondo una sequenzialità precisa.

Un modello *Swarm* è caratterizzato da una struttura gerarchica costituita su due livelli: l'*Observer*, che rappresenta il livello più esterno, appunto, dell'osservatore ed il *Model*, che costituisce il livello del modello vero e proprio.

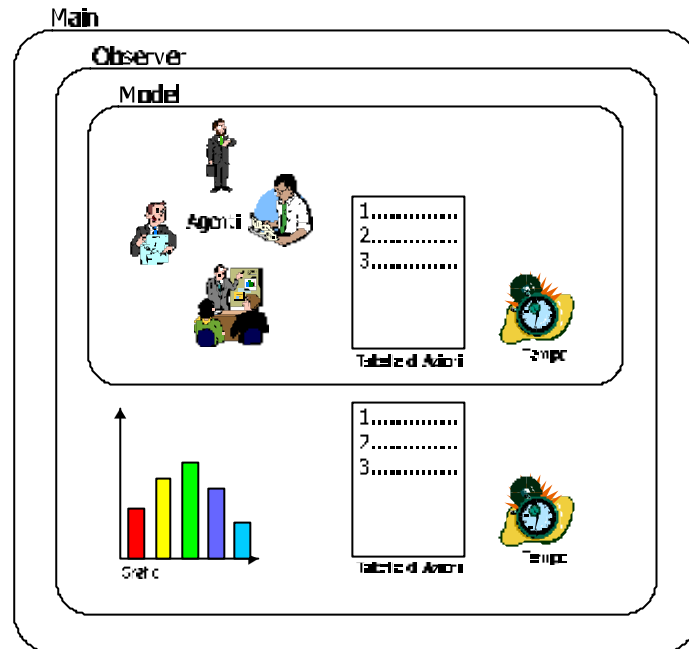
Ad un livello superiore rispetto all'*Observer* ed al *Model* si trova il programma principale, il *Main*, all'interno del quale viene creato lo stesso *Observer* e vengono impartiti i comandi attinenti agli aspetti generali del programma, come l'attivazione del *debug* o l'innescio dei numeri casuali. Il *Model* e l'*Observer*, che si possono rappresentare come l'uno dentro l'altro, possono avere alcuni elementi in comune, essenzialmente questi elementi sono oggetti, elenchi o tabelle di azioni ed il sistema di gestione del tempo, lo *schedule*.

Ad un livello inferiore, all'interno del *Model*, si trovano gli oggetti veri e propri, cioè gli agenti e tutti gli oggetti funzionali al modello, che vengono creati direttamente applicando i contenuti delle librerie di *Swarm*.

Data la struttura "nidificata" di *Swarm*, ciascun oggetto può contenerne altri organizzati gerarchicamente esattamente come il programma principale, cioè ogni oggetto può, a sua volta, contenere un intero esperimento con un proprio *Observer* ed un *Model* con la sua popolazione di agenti.

La possibilità di creare questo tipo di gerarchia, teoricamente all'infinito, fa di *Swarm* un prodotto per la simulazione completo e che permette di esplorare un enorme numero di comportamenti, permettendo, inoltre, di raffinare molto le tipologie di agenti e di renderli estremamente eterogenei e di giungere a simulare in maniera particolarmente efficace i

fenomeni complessi, che possono essere solo marginalmente e con scarsa efficacia trattati con i metodi matematico-statistici tradizionali.



**Fig.4.1** - Struttura di un modello in *Swarm*.

Nel linguaggio originario di *Swarm*, l'*Objective-C*, che orasi sta tentando di sostituire con *Java*, il codice di ogni oggetto è suddiviso in due *file*, un *header*, o "interfaccia", che contiene l'elenco dei metodi dell'oggetto ed ha estensione ".h", ed un' "implementazione", che contiene il codice vero e proprio di ogni oggetto ed ha estensione ".m", solo al momento della compilazione del programma questi due tipi di *file* danno origine ad un terzo tipo, l'oggetto vero e proprio, contraddistinto dall'estensione ".o", al termine della compilazione, inoltre, viene creato il *file* eseguibile, cioè il programma che permette di eseguire il modello.

#### **4.1.2.1 - Creazione degli oggetti.**

La prima fase di creazione degli oggetti è rappresentata dalla creazione dell'*Observer* da parte del *Main*, a sua volta l'*Observer* crea altri oggetti, cioè il *Model* e tutti gli oggetti grafici necessari a visualizzare i dati rilevanti per l'esperimento, la finestra in cui vengono inseriti e modificati i parametri di visualizzazione di ogni esperimento ed un "pannello di controllo" attraverso il quale lo sperimentatore può interagire con lo svolgimento dell'esperimento, anche solo banalmente per arrestarlo in corrispondenza di un risultato rilevante. Il *Model* crea, a sua volta, la propria finestra dei parametri significativi, attraverso i quali si può intervenire sulla

struttura del modello vero e proprio e sulla popolazione di agenti, gli agenti e gli eventuali oggetti "di servizio".

Una volta terminata la creazione degli oggetti, il *Main* ordina all'*Observer* di creare le azioni, vale a dire di creare una tabella in base alla quale scandire la rilevazione dei dati e l'aggiornamento dei grafici da effettuare durante l'esecuzione del modello. A sua volta l'*Observer* impartisce lo stesso ordine al *Model*, che crea un'analoga tabella non di rilevazione dei dati, ma di azioni, secondo una scansione temporale determinata dal proprio gestore del tempo. La sincronia fra *Model* ed *Observer* è gestito automaticamente da *Swarm*.

L'ultimo passo è l'ordine di operare del *Main* all'*Observer* e da questo al *Model*, che inizia ad eseguire tutte le sue azioni, invia cioè gli ordini agli agenti ed a tutti gli altri oggetti del modello. Il *Model* e l'*Observer* eseguono le proprie azioni in modo ciclico e sequenziale, secondo la sequenza impostata all'interno del codice e con il coordinamento di *Swarm*, che provvede a far agire i due secondo la corretta sincronia. L'esecuzione del modello, inoltre, può essere fermata e ripresa, a discrezione dello sperimentatore, grazie all'interazione con il modello stesso, che avviene grazie al "Pannello di Controllo".

Le azioni relative alla visualizzazione dei dati ed agli ordini da impartire agli agenti, vengono definite dallo sperimentatore, nel momento in cui scrive il codice; per agevolare la codifica delle interazioni fra gli agenti, lo sperimentatore può creare degli oggetti, *liste*, in cui vengono copiati gli indirizzi degli agenti. Le liste svolgono la funzione di intermediari fra il *Model* e gli oggetti, infatti, grazie all'uso delle liste, è possibile che l'invio di un ordine a tutti gli agenti possa essere fatto semplicemente mandando l'ordine alla lista, la quale provvederà a propagarlo ai vari oggetti. Per evitare, inoltre, di utilizzare sempre la stessa sequenza di attivazione, il contenuto delle liste può essere rimescolato, grazie allo *Shuffler*, un oggetto di servizio che si trova nelle librerie di *Swarm*.

I grafici di *Swarm*, infine, vengono aggiornati dall'*Observer* durante l'elaborazione, secondo una sequenza stabilita durante la scrittura del codice e modificabile, di volta in volta, dallo sperimentatore, che, inoltre, può modificare in modo dinamico il livello di dettaglio del grafico. Attraverso i grafici di *Swarm*, poi, è possibile accedere direttamente alle variabili interne dell'agente per valutarne il comportamento individuale durante lo svolgimento del modello.

L'onere di programmazione, in relazione alla completezza del risultato ottenibile, è abbastanza ridotto, infatti le variazioni da apportare al codice del *Main*, dell'*Observer* e del *Model* non sono molte, e la loro struttura è facilmente desumibile dagli "esempi" di codice che si trovano direttamente in *Swarm*; allo sperimentatore resta il compito di creare gli agenti e gli altri oggetti necessari al suo modello, di elencare le azioni da compiere, specificando, ove necessario, i tempi in cui le azioni devono essere compiute e, per ultimo, di codificare i metodi degli agenti.

Per facilitare ulteriormente il compito di chi si accinge a scrivere o modificare un programma *Swarm*, si può formalizzare uno schema, suddiviso in diversi ordini, in modo tale che gli elementi su cui effettuare eventuali modifiche siano facilmente individuabili, sia dall'autore del

codice che, a distanza di tempo, potrebbe avere difficoltà a ritrovare tutti gli elementi che compongono i diversi oggetti, sia da uno sperimentatore che voglia adattare il codice esistente ad un suo particolare lavoro.

#### **4.1.2.2 – Lo schema ERA.**

Uno dei problemi più sentiti nella sperimentazione è la dipendenza della validità dei risultati e della loro confrontabilità dall'uniformità del metodo utilizzato per raggiungerli.

L'adozione di Swarm e la conseguente scelta, in un primo momento obbligata, dell'*Objective-C* come linguaggio di programmazione, permettono di superare un primo scoglio in questo senso, ma, per quanto riguarda la struttura d'insieme del modello, non è ancora stabilita alcuna convenzione di uniformità, pur restando aperte alcune strade in tal senso.

In Terna (1998) si propone l'adozione dello schema "*Environment, Rules, Agent* (ERA)" per rendere il più possibile uniforme la struttura dei modelli ad agenti costruiti in Swarm, lo scopo dell'uso di questo tipo di struttura è certamente quello di semplificare da un lato l'uso del modello stesso da parte di ricercatori diversi e, da un altro lato, quello di rendere agevoli le modifiche effettuate anche a distanza di molto tempo dalla prima creazione del modello.

Lo schema prevede una netta separazione fra l'ambiente in cui gli agenti interagiscono e gli agenti stessi che, per gestire le proprie regole di comportamento si avvalgono di altri oggetti, specializzati rispettivamente nella produzione e nella gestione delle regole.

Secondo questo schema gli agenti non comunicano direttamente fra loro, bensì lo fanno per mezzo dell'ambiente che, in ogni caso, può fornire a ciascuno le informazioni necessarie alla conoscenza degli altri agenti presenti.

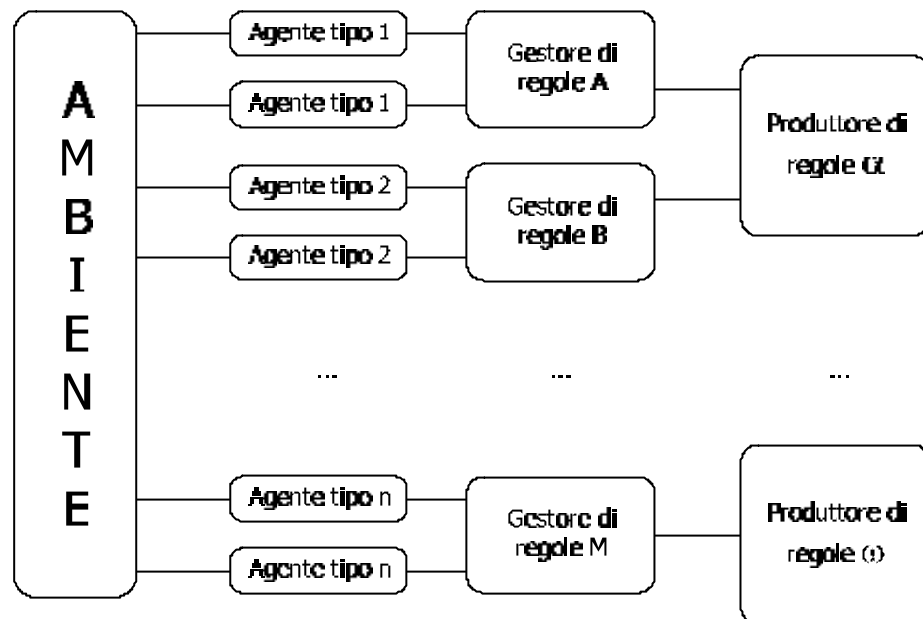
Il gestore di regole provvede a selezionare il comportamento più adatto ad una specifica situazione; quest'ultima viene descritta dalle informazioni che il singolo agente ricava dall'ambiente e riferisce al proprio gestore di regole.

Ad un livello successivo si trova il produttore di regole, il cui compito è quello di generare nuove regole, generalmente ottenute da quelle esistenti, attivando una sorta di apprendimento dell'agente.

Sull'architettura di base, è possibile inserire due ulteriori elementi, che vengono posizionati ad un livello intermedio fra l'agente ed il gestore di regole e che hanno lo scopo di facilitarne la comunicazione, sono il magazzino, *datawarehouse*, che contiene il patrimonio di regole dell'agente, ed un interprete, *interface*, che trasforma le regole dalla metrica dell'agente a quella del gestore di regole e viceversa.

L'utilizzo del *datawarehouse* permette a più agenti di condividere lo stesso gestore e lo stesso produttore di regole, pur mantenendo la totale indipendenza comportamentale gli uni dagli altri e sviluppando un proprio cammino di apprendimento; infatti il produttore ed il gestore di regole agiscono sul *datawarehouse* e l'elaborazione che viene effettuata in risposta delle

richieste di ogni singolo agente, risulta indipendente dalle precedenti e non influenza quelle che seguono.



**Fig.4.2** - Schema Environment-Rule-Agent (ERA)

Grazie all'*interface*, invece, è possibile semplificare molto il codice che descrive l'agente e le sue azioni, rendendolo al tempo stesso maggiormente indipendente dal tipo di gestore di regole utilizzato.

La suddivisione in livelli prevista dallo schema ERA, permette di individuare con facilità la posizione in cui inserire eventuali modifiche del codice, evitando al tempo stesso che, correzioni avvenute in momenti diversi, siano difficilmente individuabili.

La rigorosa struttura di questo schema evita, inoltre la duplicazione di alcune istruzioni e la conseguente moltiplicazione delle righe di codice necessarie allo svolgimento di un'azione, infatti, i vari oggetti sono specializzati per svolgere determinate funzioni e contengono tutte le istruzioni necessarie a tale scopo.

Sempre in Terna (1998): *"A second advantage of using the ERA structure is its modularity, which allows model builders to modify only the Rule Master and Rule Maker modules or objects whenever one wants to switch from agent based on a classifier system to alternatives such as neural networks, production systems or genetic algorithms."*

La modularità dello schema ERA, infatti, permette di modificare completamente il tipo di agente, semplicemente andando a modificarne il produttore ed il gestore delle regole. In questo modo diventa anche particolarmente agevole creare tipologie di agenti diversi, semplicemente aumentando i produttori e gestori di regole ed assegnando a ciascuno degli agenti, senza però



modificare la codifica degli agenti stessi. In questo modo si rende possibile il confronto di esperimenti diversi, senza la necessità di riscrivere completamente il codice.

La separazione fra l'agente ed il proprio patrimonio di regole consente, inoltre, di creare oggetti che sono in grado di dialogare con più gestori di regole, semplicemente dotando ciascun agente di più *datawarehouse* e di *interface* separate, specializzati ciascuno per ogni singolo gestore di regole.

#### **4.1.3 – La programmazione ad oggetti.**

L'utilizzo di *Swarm*, interamente scritto in *Objective-C*, ha portato almeno in un primo momento ad adottare il medesimo linguaggio di programmazione anche per la redazione dei programmi delle applicazioni, anche se nell'ultima versione *l'Objective-C* è stato quasi del tutto sostituito da Java.

*Objective-C* è un linguaggio di programmazione ad oggetti, cioè un linguaggio in cui i dati e le azioni per trattarli sono legati in un'unica entità. Gli oggetti sono a loro volta in grado di comunicare fra loro attraverso "messaggi", che permettono, all'interno di un programma, lo sviluppo di una fitta rete di interazioni fra oggetti con funzioni diverse e riducono, rispetto alla programmazione tradizionale, di ridurre il numero di righe di codice, rendendo al tempo stesso molto più chiara la logica del programma stesso.

In *Object-Oriented Programming and the Objective-C Language* (1999) *"Every object has both state (data) and behavior (operations on data). In that, they're not much different from ordinary physical objects. It's easy to see how a mechanical device, such as a pocket watch or a piano, embodies both state and behavior. But almost anything that's designed to do a job does too. Even simple things with no moving parts such as an ordinary bottle combine state (how full the bottle is, whether or not it's open, how warm its contents are) with behavior (the ability to dispense its contents at various flow rates, to be opened or closed, to withstand high or low temperatures). It's this resemblance to real things that gives objects much of their power and appeal.*

*They can not only model components of real systems, but equally as well fulfill assigned roles as components in software systems."*

Nel linguaggio *Objective-C* gli oggetti veri e propri sono creati come *instances* di una classe, di cui ereditano la struttura, ma dalla quale possono differenziarsi per ciò che concerne il valore dei dati contenuti, convenzionalmente, un oggetto e la classe cui esso appartiene sono indicati con lo stesso nome, il primo con l'iniziale minuscola e la seconda con la maiuscola. Tale convenzione è adottata anche per la redazione di *Swarm* e delle sue applicazioni.

L'attività di programmazione necessaria per la redazione di un programma *Objective-C*, consiste soprattutto nella creazione delle classi, mentre i singoli oggetti, dopo essere stati definiti, possono essere creati anche durante l'elaborazione del modello, nella quantità richiesta dalla simulazione.

Le classi sono collocate in una precisa gerarchia, il cui vertice è costituito da una classe principale, *root*, che assume un nome diverso a seconda del linguaggio ad oggetti utilizzato, grazie a questo sistema gerarchico, ogni classe dipende dalla superiore, al limite dalla *root*, e ne eredita automaticamente tutte le caratteristiche.

Il meccanismo dell'ereditarietà è fondamentale per la comprensione di un codice scritto in *Objective-C*, perché, grazie ad esso, un oggetto creato come esempio di una classe è automaticamente in grado di effettuare tutte le operazioni codificate, oltre che della classe stessa, di cui rappresenta un'*instance*, anche di tutte le classi da cui quest'ultima ha ereditato.

In questo modo il codice di ciascuna classe può essere facilmente riutilizzato e modificato *ad hoc* per ogni singolo oggetto, evitando delle inutili duplicazioni del codice e garantendo una maggiore semplicità di lettura dei programmi.

## Bibliografia Capitolo Quarto.

- AXTELL R. (2000), *Why Agents? On the Varied Motivations for Agents Computing in the Social Sciences*, reperibile all'url: <http://www.book.edu/es/dynamics>.
- BONABEAU E., DORIGO M., THERAULAZ G., (1999), *Swarm Intelligence: from Natural to Artificial Systems*, Oxford University Press.
- EPSTEIN J.M., AXTELL R. (1996), *Growing Artificial Societies – Social Science from the Bottom Up*, Ma, Mit press.
- JOHNSON P., LANCASTER A., STEFANSSON B. (1999), *Swarm User Guide*, Santa Fe Institute, New Mexico.
- MINAR N., BURKHART R., LANGTON C., ASKENAZI M. (1996), *The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations*, reperibile all'url: <http://www.santafe.edu/projects/swarm/>
- RUST J. (1996), *Dealing with the Complexity of Economic Calculations*. Working Paper. Department of Economics, Yale University: New Haven, Connecticut.
- TERNA P. (1998), *Creare mondi artificiali: una nota su Sugarscape e due commenti*, Sistemi Intelligenti, 3-98, pp. 489-496, Bologna, Il Mulino.
- TESTFATSION L. (1997), *Old Web Site for Agent-Based Computational Economics (ACE)*, reperibile all'url: <http://www.econ.iastate.edu>.

## Capitolo Quinto: un mercato di borsa.

Il modello *SUM*, *Surprising (Un)realistic Market*, è un modello ad agenti adattivi, che simula un mercato di borsa inizialmente molto particolare, infatti, la prima versione di *SUM* è costituita da un mercato artificiale popolato esclusivamente da agenti semplici, privi di aspettative sull'andamento del prezzo all'interno del mercato stesso, degli agenti definiti "*random*".

Il nucleo del modello è rappresentato dalla struttura computazionale del *book*, creato esattamente secondo le regole rigide dettate dalla Consob per definire il modo di operare del *book* del mercato telematico della borsa di Milano (MTA), ed il conseguente meccanismo di assegnazione degli *stock* durante la contrattazione: gli agenti inviano al *book* i propri ordini di vendita e di acquisto, con il relativo prezzo limite, dal *match* di tali ordini all'interno del *book* nasce il prezzo.

Nella creazione del modello si è scelto di evitare ogni semplificazione artificiosa che alterasse la spontanea formazione del prezzo, compreso il banditore marshalliano, che è stato utilizzato da Arthur nel modello *ASM*, il prezzo degli *stock* azionari è determinato esclusivamente dall'incontro dei prezzi di vendita e di acquisto all'interno del *book*.

Nelle versioni di *SUM* successive alla prima, si è scelto di introdurre all'interno del modello degli agenti di tipo cognitivo, in grado di imparare dalle esperienze passate e di sviluppare internamente le proprie regole di comportamento, allo scopo di rendere possibile l'osservazione degli effetti di tale introduzione, sia sulla loro ricchezza, sia sulla dinamica complessiva del mercato azionario.

### 5.1.1 – SUM:

Il nucleo intorno al quale è costruito il modello *SUM* è la struttura computazionale che riproduce esattamente il funzionamento del book computerizzato della borsa telematica di Milano.

Gli agenti del modello inviano al *book* i propri ordini di vendita e di acquisto, con i relativi prezzi limite, il *book* esegue immediatamente gli ordini per cui trova una controparte nel suo registro, mentre accoda in vendita ed in acquisto gli altri ordini, in modo tale da effettuare i match con gli ordini futuri. Il *book* viene cancellato all'inizio di ogni giornata di contrattazione.

Il primo risultato ottenuto dalla simulazione è l'emergenza di un mercato complesso dall'interazione di agenti estremamente semplificati che sono in grado di compiere solo alcuni tipi di azioni e non hanno alcuna visione sul mercato: gli agenti, infatti, conoscono solo l'ultimo prezzo eseguito, scelgono casualmente se comperare o vendere, fissano il loro prezzo limite moltiplicando l'ultimo prezzo eseguito per un coefficiente casuale.

La struttura del mercato, costituita dal *book*, fa sì che anche in presenza di agenti semplicissimi, il mercato mostri delle sequenze di prezzo crescenti e decrescenti, che mostrano una significativa volatilità. In questo mercato "elementare" compaiono anche bolle e *crash*, dovuti esclusivamente alla struttura del mercato, senza bisogno di alcuna spiegazione esogena al mercato stesso.

Nelle successive versioni del modello sono stati inseriti altri tipi di agenti, allo scopo di valutare le conseguenze dell'introduzione di agenti che utilizzano l'analisi tecnica o le reti neurali per prevedere l'andamento i prezzi futuri, di agenti cognitivi, in grado di imparare dalle proprie esperienze e di perseguire il raggiungimento di obiettivi esterni mediante lo schema dei *Cross Target (CT)*, illustrato in Terna (2000), di agenti cognitivi in grado di apprendere dalle proprie esperienze e dai risultati conseguiti nel passato grazie al metodo dei *classifier system*.

Le prossime versioni di SUM si prevedono di introdurre agenti che agiscono seguendo modelli ad aspettative razionali e l'introduzione, provocatoria, di un astrologo che prevede l'andamento del mercato.

### 5.1.2 – Agenti con e senza "mente".

In Terna (2000): *"In computer simulation models based upon agents, what is the degree of sophistication that we have to put into the agents? Should we provide them or not with a "mind"? The answer ranges from Axelrod's simplicity principle to the use of full BDI (Beliefs, Intentions, Desires) cognitive agents. To discuss the subject we introduce here three models: one with "no-mind" agents that operate in an unstructured market, the second with "minded" agents assuring some stability to an emerging unstructured market and, finally, the third with blind agents, that show a sophisticated outcome in a structured market. No generalised results come from this presentation, but many useful doubts."*

Il problema che si pone uno sperimentatore, quando progetta un modello ad agenti adattivi, riguarda essenzialmente la complessità, e da dove essa scaturisca, lo sperimentatore, all'atto dell'impostazione del modello, deve stabilire se propendere per un'impostazione maggiormente

legata al riduzionismo dell'individualismo metodologico e, quindi, includere la complessità negli agenti stessi oppure se optare per l'estrema semplicità degli agenti e lasciare che semplicemente dalla loro interazione emergano comportamenti complessi.

Entrambi i punti di vista comportano degli inconvenienti, infatti, seguendo la prima via si rischia di legare il modello a condizioni iniziali arbitrariamente definite e di incorrere in una relazione di autoreferenza, mentre con la seconda via, probabilmente, non si riesce a far convergere il modello ad alcuna soluzione plausibilmente fondata, ma si genera semplicemente del "rumore bianco".

La domanda che ci si deve porre è: se la simulazione è basata su agenti, quanto devono essere complicate le loro "capacità intellettive"? E' necessario fornirli di una "mente"?

Una risposta potrebbe essere quella rifiutata da Axelrod (1997), e che propende per la creazione di agenti cognitivi dotati di convinzioni, intenzioni e desideri (*BDI*), ma la via che sembra garantire i migliori risultati è quella intermedia, procedere con agenti ragionevolmente semplici, secondo il principio KISS (Keep It Simple, Stupid) di Axelrod, ricercando nell'interazione e nella presenza di strutture ex-ante i mezzi adatti a riprodurre la reale complessità, che potrà essere interpretata secondo la scala migliore, non necessariamente coincidente con quella individuale.

In Terna (2000), prendendo in considerazione, oltre alle capacità cognitive degli agenti, anche la struttura dell'ambiente circostante, si riscontano quattro tipi di possibili combinazioni:

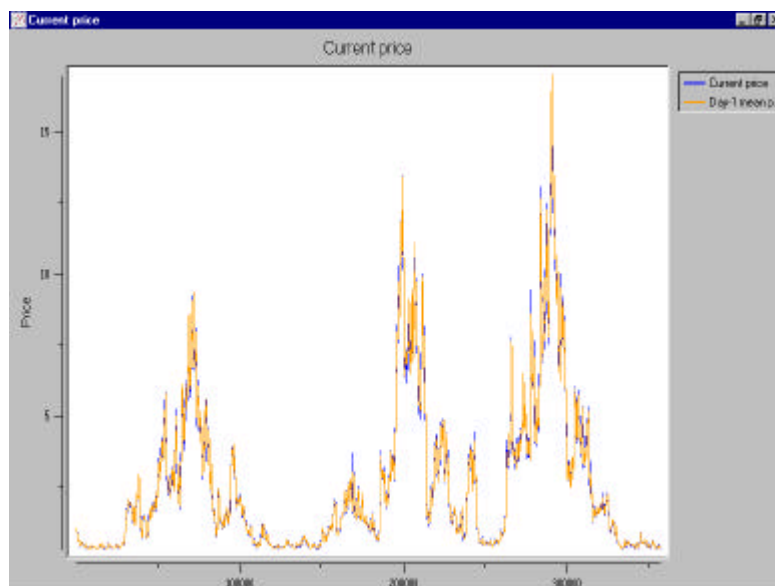
- 1 - Agenti "senza mente", che agiscono in un ambiente non strutturato.
- 2 - Agenti cognitivi in un ambiente non strutturato.
- 3 - Agenti "senza mente" in un ambiente strutturato.
- 4 - Agenti cognitivi in ambiente strutturato.

Il modello SUM, mostra le dinamiche della formazione delle serie storiche del prezzo in un mercato di borsa, analizzando in maniera approfondita i casi 3 e 4, cioè quelli che prevedono agenti semplici, privi di "capacità intellettive, ma che operano in un ambiente dotato di struttura ed agenti cognitivi, che operano nello stesso ambiente strutturato.

I precedenti casi 1 e 2, infatti, generano degli andamenti caotici di scarso significato economico, definibile essenzialmente come "rumore bianco"; negli altri casi e, in particolare modo, nel caso di agenti "senza mente" in un mercato strutturato, è molto interessante vedere come le serie storiche del prezzo di borsa è in grado di generare bolle e crash.

L'esempio di grafico che segue è relativo proprio a questo tipo di mercato, infatti si riferisce ad una simulazione del modello SUM, con 200 agenti di tipo *randomAgent*, cioè agenti *no mind*, che operano, esclusivamente coordinati dalla struttura del mercato di borsa, cioè la struttura del mercato telematico di Milano, dettata dalla Consob.

La serie dei prezzi presenta, come è evidente dal grafico, delle bolle e dei *crash*, che non possono essere spiegati da elementi esogeni alla struttura dell'ambiente in cui gli agenti operano.



**Fig.5.1** – Grafico del prezzo in un mercato strutturato con agenti *no mind*.

Il mercato che emerge da questa simulazione risulta sorprendentemente realistico, dal momento che un'attenta lettura della serie dei prezzi, se il modello viene fatto girare sufficientemente a lungo, evidenzia un mercato abbastanza stabile e poco movimentato, in cui si evidenziano pochissime bolle in un arco di tempo artificiale rapportabile ad un secolo di tempo reale.

L'aspetto sorprendente deriva dal fatto che gli agenti *random* non hanno alcuna visione complessiva del mercato e che, tuttavia, si possono ravvisare dei parallelismi con la storia borsistica mondiale dell'ultimo secolo, infatti, nell'arco di tempo corrispondente ad un secolo, si formano solo due bolle di grande entità, con i successivi crolli, proprio come è avvenuto nel secolo scorso per le due grandi crisi borsistiche del 1929 e del 1987.

Nelle successive versioni del modello, di cui si analizzeranno in seguito i risultati nel dettaglio, sono stati introdotti agenti cognitivi, in grado di conoscere il mercato, di formulare previsioni sull'andamento del prezzo delle azioni e di adeguare il proprio comportamento alla previsione, in questo modo tratteremo il secondo caso annunciato, gli agenti cognitivi che operano in un mercato strutturato ed avremo la possibilità di valutare separatamente sia gli effetti che la struttura ha sul comportamento degli agenti, sia quella degli agenti cognitivi sull'andamento del mercato.

### **5.1.3 – Il book ed il meccanismo di formazione del prezzo.**

La forma dell'oggetto "*book*" del modello corrisponde esattamente a quella dettata dalla Consob per il mercato telematico azionario della borsa di Milano.

Il mercato telematico azionario (MTA) indica il comparto di mercato in cui si negoziano azioni, obbligazioni convertibili, warrant, diritti di opzioni e quote di fondi mobiliari ed immobiliari chiusi quotati in borsa.

Secondo il regolamento di Borsaitalia S.p.a., vi sono diverse fasi che riguardano la negoziazione di un titolo, in primo luogo, c'è l'asta di apertura, durante la quale si determina il prezzo teorico di apertura (pre-apertura), segue la fase di validazione del prezzo teorico di apertura (validazione), ed infine la fase in cui si concludono i contratti (apertura), l'ultima fase è costituita dalla negoziazione continua, introdotta con l'avvento del mercato telematico.

La volontà negoziale degli operatori viene espressa mediante proposte di negoziazione, che contengono informazioni relative allo strumento finanziario da negoziare, alla quantità, al tipo di operazione, al tipo di conto ed alle eventuali condizioni sul prezzo; nel mercato telematico italiano, tutte le operazioni di immissione, modifica e cancellazione degli ordini possono essere ugualmente effettuate sia nella fase di pre-apertura, sia in quella di contrattazione continua.

Uno degli aspetti fondamentali per la struttura di SUM è il modo in cui il book telematico gestisce gli ordini di acquisto e di vendita, una volta che questi vi sono stati immessi dagli operatori, infatti, proprio dalla struttura interna del book dipendono i meccanismi che determinano i prezzi delle negoziazioni.

Le proposte di vendita e di acquisto sono automaticamente ordinate nel mercato per ciascuno strumento finanziario in base al prezzo – decrescente se in acquisto e crescente se in vendita – nonché, a parità di prezzo, in base alla priorità temporale determinata dall'orario di immissione della proposta, inoltre, le proposte modificate perdono la priorità temporale acquisita, se la modifica implica un aumento della quantità o una variazione del prezzo di riserva.

Le proposte di acquisto o vendita di azioni, obbligazioni convertibili e warrant possono avere per oggetto quantitativi pari o multipli del lotto minimo di negoziazione (proposte "intere") o quantitativi minori del lotto minimo di negoziazione ("spezzature").

Il *book* di SUM, pur mantenendo l'impostazione generale del *book* ufficiale di Borsaitalia S.p.a., è stato semplificato sotto alcuni aspetti, funzionali rispetto allo sviluppo del modello; in modo particolare, per quel che riguarda la fase di pre-apertura, infatti, non si determina il prezzo di apertura sulla base della pre-apertura, ma si considera come informazione il prezzo ultimo della giornata precedente, pertanto gli ordini immessi in pre-apertura non sono conclusi a un prezzo definito, ma sono passati alla negoziazione continua; tali ordini, immessi dagli operatori con una probabilità pari al 5% (eventualmente modificabile), servono solo a iniziare con un *book* non vuoto, per evitare anomalie di prezzo ad ogni inizio di giornata.

Inoltre, le proposte immesse in pre-apertura non sono trattate fra loro, ma troveranno contropartita solo nel durante, motivo per cui durante la pre-apertura non si determina alcun prezzo e non vengono effettuate le fasi di validazione e quella di apertura vera e propria.

Nella fase del cosiddetto "Durante", le proposte sono ordinate secondo il prezzo: in modo decrescente se in acquisto ed in modo crescente se in vendita; a parità di prezzo, come nel *book* ufficiale, sono ordinate per priorità di tempo; inoltre, le proposte possono essere immesse con limite di prezzo o senza limite di prezzo, mentre esiste una limitazione sulla quantità, in quanto tutti trattano il lotto minimo, anche se ogni agente può immettere più ordini.



Dal punto di vista dell'operatività, il *match* degli ordini avviene in questo modo: l'immissione di una proposta con limite di prezzo in acquisto determina l'abbinamento con una proposta di vendita avente prezzo inferiore o uguale a quello della proposta immessa, mentre dal versante opposto, l'immissione di una proposta con limite di prezzo in vendita determina l'abbinamento con una proposta di acquisto avente prezzo superiore o uguale a quello della proposta immessa.

La determinazione del prezzo di conclusione del contratto di compra-vendita prevede che, per ogni contratto concluso mediante abbinamento automatico ai sensi delle precedenti regole, il prezzo è pari a quello della proposta avente priorità temporale superiore.

#### 5.1.4 – I parametri del modello SUM.

I parametri dell'ultima versione di SUM, la versione 0.65, per quanto riguarda l'*ObserverSwarm*, cioè la struttura di *Swarm* che permette di rappresentare graficamente i risultati, sono stati inseriti o modificati i seguenti parametri:

- *displayFrequency*, è la frequenza con cui l'*Observer* traccia il grafico degli eventi del modello, nell'ambito di SUM, si è optato per una frequenza di visualizzazione uguale o multipla del numero degli agenti del modello, in modo tale che ad ogni "tick" del grafico corrisponda un giorno di contrattazioni, infatti ogni agente opera una sola volta ogni giorno.
- *stopAtEpochNumber*, indica il numero di giorni, trascorsi i quali la simulazione si arresta, il minimo perché almeno un agente possa operare è un giorno.

Parametri del *ModelSwarm*, sono i parametri strutturali del modello, cioè quelli che coordinano le azioni degli agenti:

- *agentNumber*, indica il numero degli agenti che operano nel modello ogni giorno, questo valore è calcolato internamente e corrisponde alla somma di tutti gli agenti di ogni tipo, che sono presenti in ogni singola simulazione;
- *randomAgentNumber*, corrisponde al numero di agenti che agiscono in maniera casuale, cioè gli agenti più semplici, precedentemente definiti *no mind*;
- *marketImitatingAgentNumber*, è il numero degli agenti che decidono di comperare o vendere imitando l'andamento del mercato, scelgono di vendere se il prezzo medio negli ultimi due giorni è sceso e scelgono di acquistare se negli ultimi due giorni il prezzo è salito;

- *locallyImitativAgentNumber*, sono gli agenti che scelgono se vendere o comperare sulla base di ciò che ha fatto la maggioranza degli ultimi N agenti (l'ampiezza della finestra è determinata dal parametro *localHistoryLenght*);
- *stopLossAgentNumber*, sono gli agenti che comperano o vendono per limitare le perdite, infatti lo "stop loss" è il prezzo inferiore al proprio livello d'acquisto, che, se raggiunto farà chiudere necessariamente all'agente la propria posizione con la perdita massima che si era proposto di accettare;
- *aNNForecastAppAgentNumber*, indica il numero degli agenti che seguono le previsioni effettuate da un agente previsore, comperando o vendendo in base ai prezzi futuri stimati per mezzo di una rete neurale, agendo anche allo scoperto;
- *bPCTAgentANumber*, è il numero degli agenti che utilizzano il metodo dei *Cross-Target* e corrispondono al tipo A, che verrà analizzato in seguito nel dettaglio;
- *bPCTAgentBNumber*, è il numero degli agenti che utilizzano il metodo dei *Cross-Target* e corrispondono al tipo B, che verrà analizzato in seguito nel dettaglio;

Parametri generali del modello:

- *asymmetricBuySellProb*, è la probabilità che gli agenti imitatori locali o del mercato scelgano di seguire la maggioranza o di compiere l'azione opposta; gli agenti imitatori hanno una probabilità pari a  $p$  di imitare la maggioranza e probabilità pari a  $(1-p)$  di non imitarla; (gli agenti *random* hanno la stessa probabilità di acquistare o vendere, infatti,  $p = 0.5$ );
- *agentProbToActBeforeOpening*, per il meccanismo del *book*, indica la probabilità di inserire una proposta di ordine nella fase di pre-apertura, all'interno del modello serve esclusivamente per iniziare la giornata con una coda di ordini nel *book*. Tutti i tipi di agenti del modello hanno la possibilità di agire in pre-apertura;
- *floorP*, indica un prezzo limite, sotto il quale gli agenti possono comunque scegliere di operare con una probabilità determinata da un parametro che si può modificare dal *Model*;
- *agentProbToActBelowFloorPrice*, indica la probabilità che un agente decida di acquistare nel caso il prezzo scenda al di sotto del *floorPrice*;

- *maxOrderQuantity*, indica la quantità massima di azioni che ogni agente può acquistare o vendere per ogni ordine, nel modello questo parametro è costituito da un numero intero scelto casualmente fra 1 e *maxOrderQuantity*;

Parametri caratteristici degli agenti imitatori:

- *meanPriceHistoryLenght*, questo parametro è costituito da un vettore, i cui elementi sono i prezzi medi, questo parametro è anche utilizzato dal previsore neurale per effettuare la previsione;
- *localHistoryLenght*, questo parametro è costituito da un vettore che memorizza le azioni degli agenti (permette agli imitatori di capire se la maggioranza degli agenti ha acquistato o venduto);

Parametri degli agenti di tipo *stop loss*:

- *maxLossRate*, indica il limite di perdita massimo, oltre il quale l'agente decide comunque di operare per limitare le perdite;
- *stopLossInterval*, indica l'intervallo di attuazione dello stop loss;
- *checkingIfShortOrLong*, serve all'agente per capire se in un dato momento è *Short* o *Long* e, quindi, decidere se deve acquistare o vendere;

Parametri utilizzati dall'agente che effettua previsioni con la rete neurale:

- *dataWindowLenght*, indica il numero dei dati utilizzati come *input* della rete neurale artificiale (ANN),
- *nAheadForecasting*, indica il giorno a cui si riverisce la previsione della rete neurale, viene indicato come n giorni avanti al momento in cui la previsione è effettuata;
- *forecastingTrainingSetLenght*, indica il numero di input e di output attesi utilizzati per l'apprendimento della rete neurale, in questo specifico caso si tratta di un *training set* mobile, che viene modificato con il progredire del tempo interno all'esperimento;
- *epochNumberInEachForecastingTrainingCycle*, indica il numero delle epoche di apprendimento in ciascuna fase di training della rete neurale;

- *learningProcessEveryNDays*, è il parametro che indica ogni quanti giorni la rete neurale ripete l'apprendimento;
- *cleanForecastingANNEveryMgtemNDays*, ogni M giorni (con M multiplo o uguale ad N), il processo di apprendimento ricomincia dall'inizio con i parametri *random* iniziali;

Parametri utilizzati dagli agenti che applicano la previsione effettuata con la rete neurale:

- *aNNInactivityRange*, indica il range delle variazioni di prezzo previsto in cui questo tipo di agente non agisce;
- *aNNForecastAppAgentActDailyProb*, è la probabilità di agire ogni giorno di questo tipo di agenti;

Parametri utilizzati dagli agenti costruiti secondo lo schema BPCT:

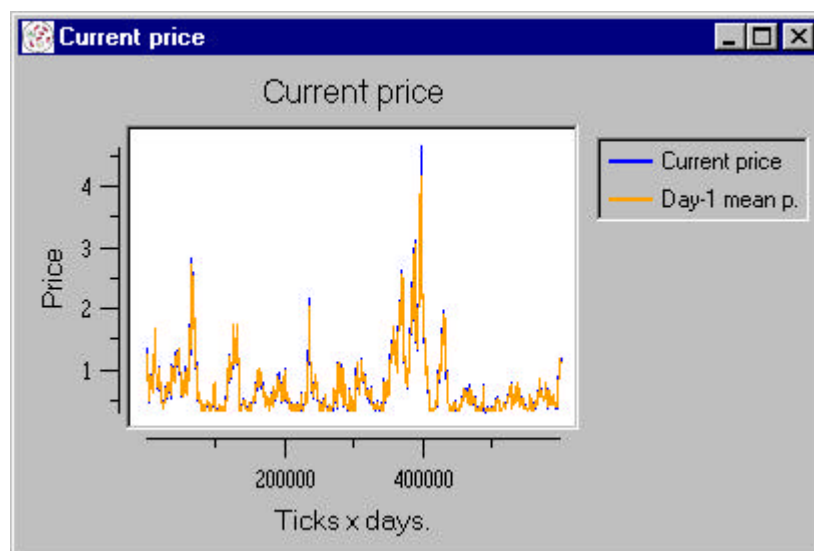
- *epochNumberInEachBPCTTrainingCycle*, l'apprendimento, nel nostro caso, ha un'*epochNumber* pari a 100 ed è calcolato ogni giorno su una finestra mobile, che prende in considerazione gli ultimi 10 giorni;
- *agentAEO\_EPDelta*, è un parametro che serve a valutare gli effetti degli Obiettivi Esterni per gli agenti CT di tipo A;
- *agentBEO\_EPDelta*, è un parametro che serve a valutare gli effetti degli Obiettivi Esterni per gli agenti CT di tipo B;

### 5.1.5 – Esperimenti solo con agenti *random*.

Nella prima applicazione del modello SUM, sono stati inseriti solo agenti *random*, il modello, dunque, corrisponde al tipo precedentemente illustrato, di agenti "senza mente" che operano in un mercato strutturato.

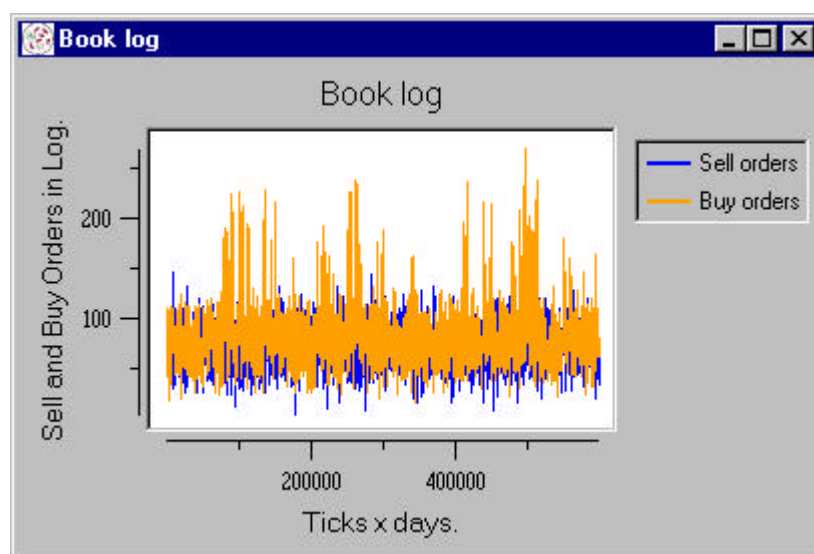
Come è stato riportato nell'illustrazione dei parametri del modello, gli agenti *random* conoscono solamente l'ultimo prezzo eseguito, scelgono casualmente se acquistare o vendere e fissano il proprio prezzo limite moltiplicando il prezzo appena eseguito per un coefficiente casuale.

Inoltre, all'interno del modello viene determinata una probabilità di fissa, in base alla quale, se il prezzo scende al di sotto del *floorPrice*, gli agenti acquistano azioni.



**Fig. 5.2-** Grafico del prezzo in presenza in un modello popolato solo da 300 agenti *random*, con un parametro di *maxOrderQuantity* pari a 3 .

Anche con agenti così semplici il grafico del prezzo evidenzia bolle speculative e crash, conseguenza della struttura del mercato (costituita dal *book* elettronico), all'interno del modello, quindi, si verifica l'emergenza di comportamenti complessi, derivanti da regole di comportamento semplici.

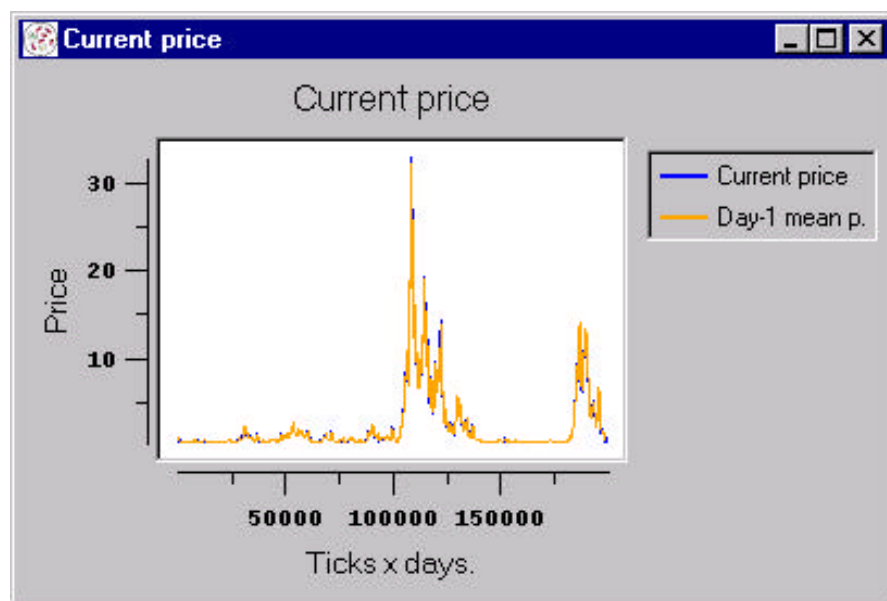


**Fig.5.3 -** Grafico del *book*, corrispondente al grafico del prezzo della Fig. 5.2.

Le regole del *book*, che rappresentano l'unica struttura tecnica del mercato, influenzano in modo determinante il modello, poiché intervengono direttamente non soltanto sui possibili comportamenti degli agenti, ma anche sui meccanismi di interazione, riuscendo, così, a far emergere risultati complessi da azioni semplici, e, soprattutto, da agenti semplici.

I prezzi sono generati dal *match* fra gli ordini di vendita e gli ordini di acquisto immessi all'interno del book da ogni agente, ogni giorno, il prezzo viene stabilito solo nel momento in cui viene eseguito, tutti i prezzi eseguiti vengono disegnati sul grafico del prezzo.

Il mercato descritto in questo modello è particolare sotto alcuni punti di vista, in primo luogo si nota che se al suo interno operano pochi agenti che hanno la possibilità di effettuare più di un ordine al giorno, emergono bolle e crash, anche rilevanti, in fatti, se si imposta un esperimento con 100 agenti, che hanno la possibilità di effettuare fino a 6 ordini, si formano bolle di entità rilevante, all'apice delle quali il prezzo raggiunge quasi trenta volte il prezzo iniziale.



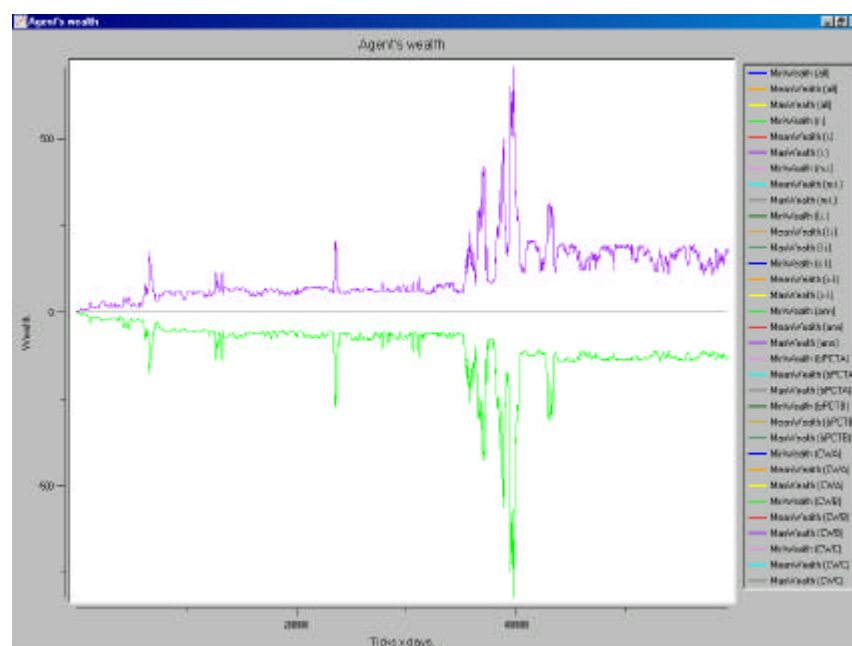
**Fig. 5.4** – Grafico del prezzo con 100 agenti e *maxOrderQuantity* = 6.

La prima interpretazione delle cause delle bolle speculative, deriva da un'analisi del *log* del *book*, infatti si può notare che nel momento in cui le bolle nascono si hanno delle code di ordini, che possono essere anche molto lunghe, in vendita o in acquisto.

La presenza di sequenze di azioni tutte da un solo lato del mercato, che si verifica proprio nel momento in cui le bolle ed i crash si generano sembra essere l'unica spiegazione dei fenomeni stessi. Perché queste situazioni si verifichino e, soprattutto si verifichino in un mercato popolato da agenti casuali, è molto più complicato da giustificare.

Il modello, inoltre, permette di valutare la prevedibilità del mercato, grazie ad un agente "previsore", tale agente effettua una stima sul prezzo di dieci giorni avanti (10 è un parametro direttamente modificabile dal ModelSwarm), basando la propria previsione su una rete neurale artificiale, ma non opera direttamente sul mercato.

Per valutare la prevedibilità del mercato viene utilizzato un indice determinato dalla quota percentuale di segni esatti nella previsione delle variazioni dei prezzi, una previsione viene



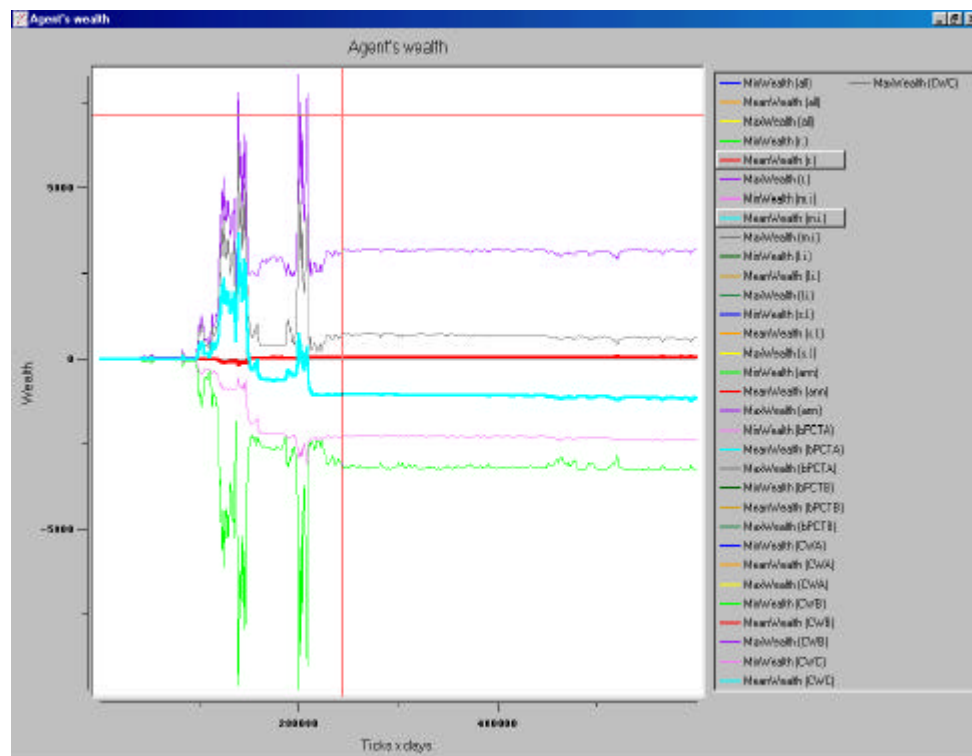
**Fig. 5.5** – Grafico del livello di ricchezza in un mercato con 300 agenti casuali e  $maxOrderQuantity = 3$ .

### 5.1.6 – Imitatori del mercato, imitatori locali ed agenti *stop loss*.

Per valutare gli effetti dell'interazione con agenti di tipo diverso, all'interno del modello sono stati introdotti altri tre tipi di agenti: degli imitatori locali, cioè agenti che possono decidere di imitare il comportamento degli  $n$  agenti che hanno operato per ultimi, degli imitatori del mercato e degli agenti che operano limitando le perdite, *stop loss*.

Gli agenti imitatori del mercato, *MarketImitatingAgents*, decidono se vendere o acquistare titoli in base ad un coefficiente di probabilità  $p = asymmetricBuySellProb$ , che determina la probabilità di comperare o vendere secondo il suggerimento del comportamento imitativo o di fare esattamente l'opposto, la probabilità è naturalmente sbilanciata verso il comportamento imitativo, ma non esclude che l'agente possa comportarsi in modo diverso.

Il parametro  $p$  misura la probabilità che un agente, del tipo *marketImitatingAgent*, scelga se acquistare o vendere imitando il mercato nel suo complesso, decidendo di acquistare con probabilità  $= p$ , se il prezzo medio è salito dal giorno (t-2) al giorno (t-1) e decidendo di vendere con probabilità  $= (1-p)$  *coeteris paribus*, al contrario, l'agente deciderà di vendere con probabilità  $p$ , se da (t -2) a (t-1) il prezzo medio è diminuito e di operare in controtendenza sempre con probabilità  $(1-p)$ .





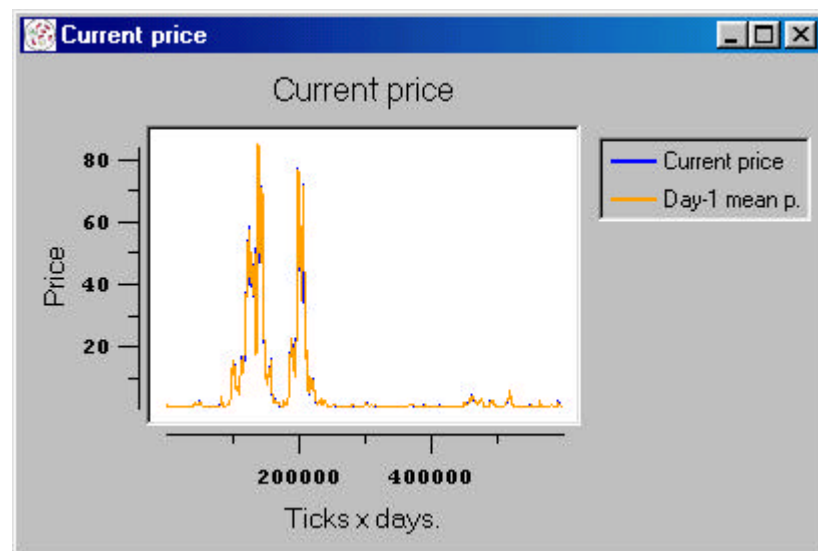
**Fig. 5.6** – Market initiating Wealth in un mercato con 285 randomAgent, 15 imitatori del mercato e *maxOrderQuantity* pari a 3.

Il comportamento di questo tipo di agenti è sì imitativo, ma si avvicina anche molto a quello dei cosiddetti *chartisti*.

La presenza di questo tipo di agenti, anche se in quantità ridotta rispetto al mercato nel suo complesso, esaspera le bolle speculative ed i crash, fino a raggiungere livelli di prezzo pari ad ottanta volte il prezzo iniziale.

Sotto il punto di vista della prevedibilità, però, la situazione migliora fino a raggiungere la quota di 53 punti percentuali di previsioni di segno corretto.

Analizzando, invece, l'effetto del comportamento degli agenti sulla loro ricchezza, si può notare l'enorme effetto che questo genere di condotta esercita sul livello delle dotazioni ed il particolare effetto indotto dall'imitazione, nel limitare le perdite nel momento dello sgonfiamento della bolla.



**Fig. 5.7** – Sequenza del prezzo in una simulazione con 285 agenti random e 15 agenti che imitano il mercato, tutti gli agenti hanno una *maxOrderQuantity*=3.

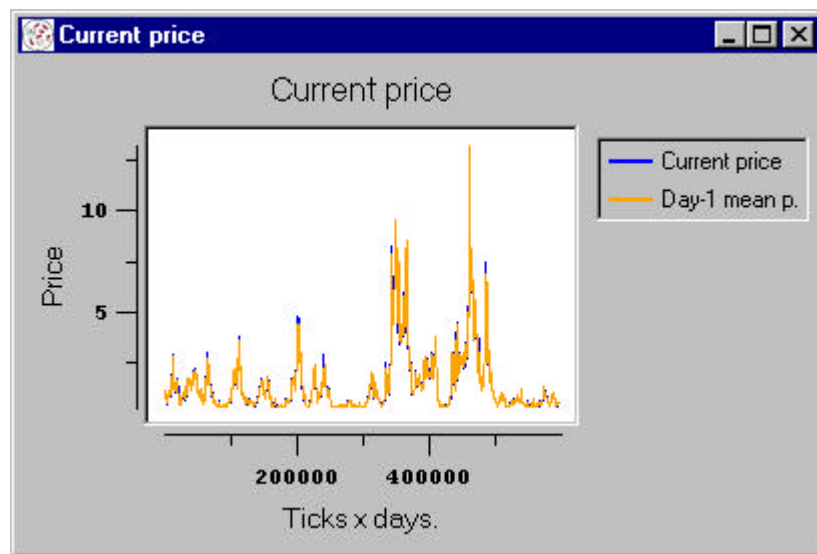
Successivamente agli imitatori del mercato, nel modello sono stati introdotti degli imitatori locali, cioè degli agenti che non hanno una visione completa del mercato, ma conoscono soltanto le azioni degli ultimi *n* agenti che hanno operato (ad esempio gli ultimi 20 agenti).

La probabilità di acquistare o vendere viene gestita, da questi agenti, secondo il parametro  $p = \text{asymmetricBuySellProb}$  già analizzato per gli imitatori di mercato, ma in questo caso tale probabilità viene applicata solo da una parte del mercato, infatti,  $p$  indica la probabilità di seguire l'esempio della maggioranza degli ultimi *n* agenti che hanno operato e che vengono individuati da

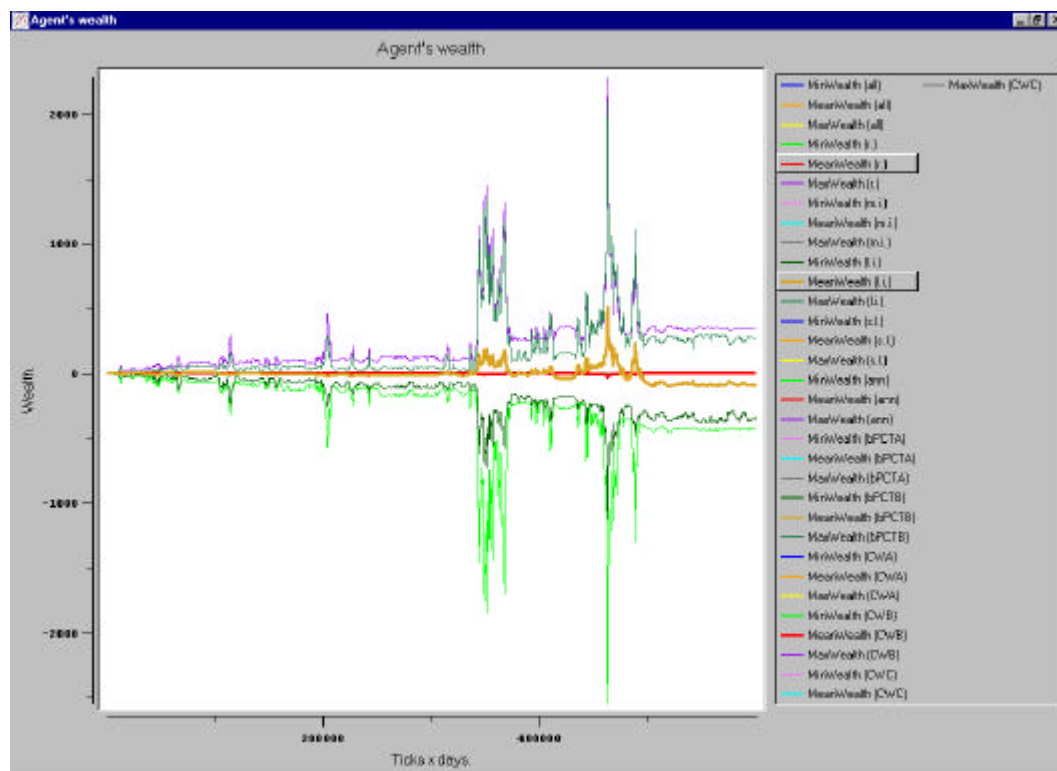
una sequenza inserita all'interno di una finestra mobile, in modo da garantire che gli agenti imitati siano sempre gli ultimi  $n$  in ordine temporale.

Il mercato che emerge presenta bolle meno esasperate rispetto a quello in cui operano agenti casuali ed imitatori del mercato, ma è anche meno prevedibile, infatti presenta solo il 47% dei segni corretti nella previsione.

Gli imitatori locali, inoltre, limitano le perdite durante gli sgonfiamenti delle bolle, facendo un confronto con i *random*, ma risentono del *crash* più degli imitatori del mercato.



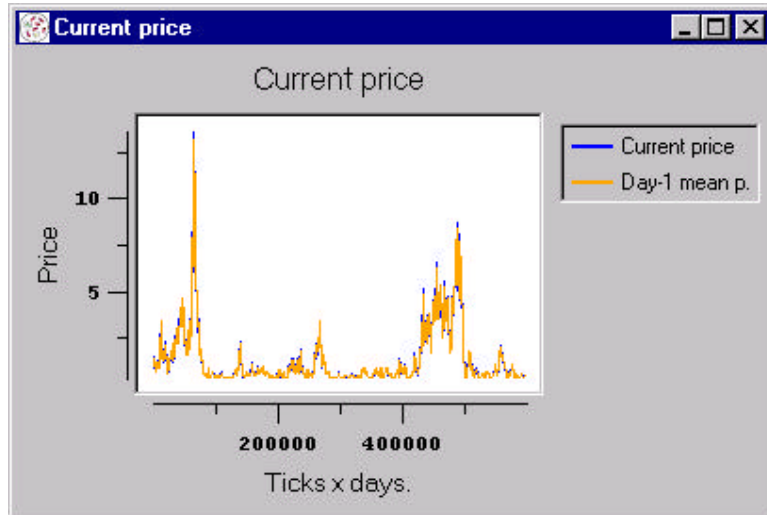
**Fig. 5.8** – Prezzo in un mercato con 285 agenti casuali, 15 imitatori locali e *maxOrderQuantity* pari a 3



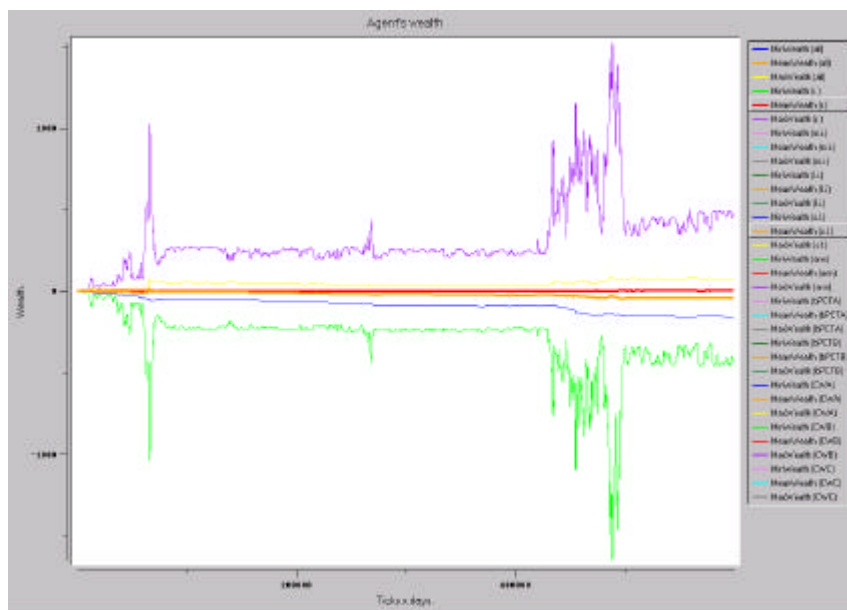
**Fig. 5.9** – Ricchezza in un mercato con 285 agenti casuali, 15 imitatori locali e *maxOrderQuantity* pari a 3

L'ultimo tipo di agenti *chartisti* è costituito dagli *stop loss*, cioè dagli agenti che operano decidendo di limitare le perdite, gli agenti *stop loss*, infatti, decidono di comperare o vendere se il prezzo corrente, confrontato con il prezzo (*t-stopLossInterval*), cresce o diminuisce ad un tasso superiore o uguale al parametro *maxLossRate*, che indica il limite di perdita massima che ciascun agente si è imposto di accettare al momento del suo ingresso sul mercato. L'introduzione degli agenti *stop loss*, all'interno del mercato artificiale, genera bolle di entità ragionevole, il livello del prezzo raggiunge circa 13 volte il prezzo iniziale, il mercato, inoltre, risulta anche sufficientemente prevedibile, infatti la quota dei segni corretti nella previsione raggiunge il 54%.

Per quanto riguarda l'effetto sulla ricchezza, gli agenti *stop loss* presentano un andamento della ricchezza prossimo allo zero, e non risentono delle bolle e dei *crash*.



**Fig.5.10** – Grafico del prezzo in un mercato con 285 randomAgent e 15 stop loss, MaxOrderQuantity pari a 3.



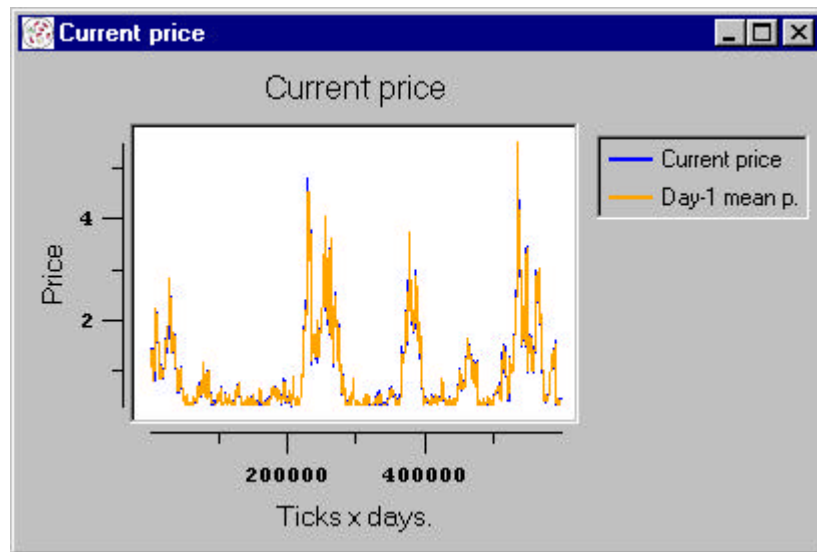
**Fig. 5.11** – Livello di ricchezza degli agenti stop loss all'interno di un mercato con 285 randomAgent, 15 stop loss con maxOrderQuantity pari a 3.

### 5.1.7 – Analisi del mercato sotto il profilo della previsione.

Nei paragrafi precedenti è stato introdotto un agente in grado di effettuare una previsione sull'andamento del mercato futuro, basando la propria previsione su una rete neurale, ma che non operava sul mercato.

In questo paragrafo, invece, vengono introdotti degli agenti che seguono la previsione del *forecaster*, cioè degli agenti che operano sul mercato seguendo le stime effettuate dall'agente previsione sull'andamento dei prezzi.





**Fig. 5.13** – Grafico del prezzo con 285 randomAgent, 15 *aNNForecastAppAgent* e maxOrderQuantity pari a 3.

L'agente previsione compie una stima del prezzo di mercato  $n$  giorni avanti basandosi su una rete neurale artificiale dotato di una finestra di trenta dati di *input*. I dati sono costruiti come indici del prezzo medio di ogni giorno sul prezzo medio giornaliero di  $n$  giorni prima.

L'output della rete neurale, invece, è costituito dalla stima  $p_n/p_0$ , nel caso dell'esperimento  $n=10$ .

Il parametro *forecastingTraining SetLenght* rappresenta il numero delle sequenze di input ed output attesi usati per testare la rete neurale, nel caso dell'esperimento il training set viene calcolato sulla base di una finestra mobile, che si modifica secondo il procedere del tempo interno all'esperimento stesso.

*EpochNumberInEachForecastingTrainingCycle* è il numero delle epoche di apprendimento in ogni fase di *training* della rete neurale; ogni epoca contiene un numero di cicli di training pari a *forecastingTrainingSetLenght*.

Il processo di training della rete neurale viene ripetuto ogni 10 giorni, 10 è una variabile che corrisponde ad un parametro *learningProcessEveryNDays*. Infine il processo di apprendimento ricomincia con i parametri random iniziali, che corrispondono ai pesi della rete neurale, per consentire maggiori cambiamenti nella struttura degli input.

Il numero dei nodi nascosti, *hidden*, è determinato internamente come la metà dei nodi di input, tuttavia questo valore può essere variato esternamente dal file *forecasting.setup*.

Sulla struttura di base di Sum0.65 vengono inseriti degli agenti che seguono il previsore neurale, che vendono ed acquistano allo scoperto, seguendo in maniera meccanica la previsione, questo tipo di agenti, cioè, acquista o vende ogni volta che la previsione è rispettivamente più alta o più bassa rispetto al prezzo corrente.

Più nel dettaglio, gli agenti non operano se l'indice di previsione è superiore al valore di  $(1+aNNInactivityRange)$  o se è inferiore a  $(1- aNNInactivityRange)$ , inoltre gli agenti che applicano la previsione sono in grado di evitare le azioni insensate, non agendo se il prezzo corrente è già superiore a quello previsto.

Se nel modello vengono introdotti pochi agenti, emerge un mercato con bolle abbastanza contenute ed una varianza ragionevole nei prezzi. In questo caso la quota di prevedibilità del mercato raggiunge il 52%.

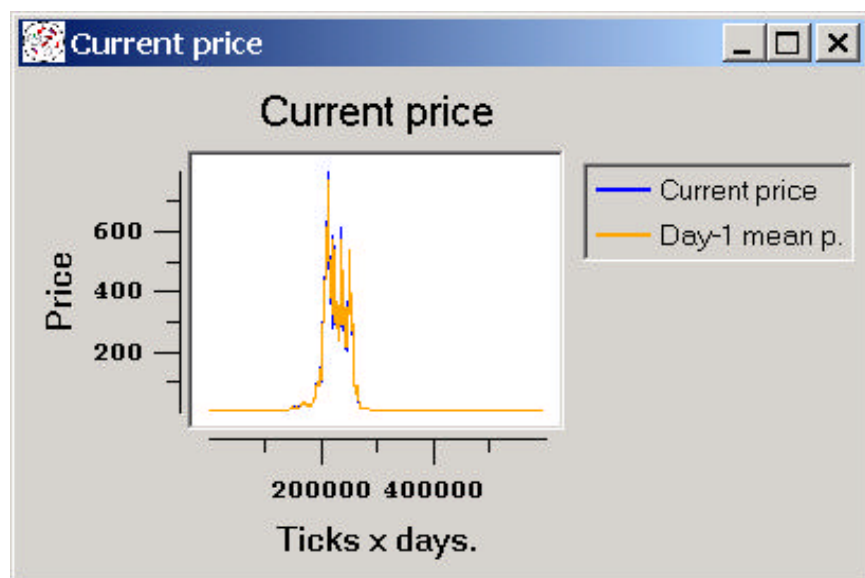


Fig. 5.14 – Mercato con 30 *aNNForecastAppAgent*, 270 *randomAgent*, *maxOrderQuantity* pari a 3.

Analizzando gli effetti del comportamento degli agenti sul proprio livello di ricchezza, si può notare che gli agenti che applicano la previsione si posizionano leggermente spostati verso il guadagno rispetto agli agenti *random*, che perdono maggiormente e risentono soprattutto delle bolle in modo più forte.

Se, però il numero degli agenti che applicano la previsione della rete neurale viene raddoppiato, cioè nel modello vengono introdotti trenta agenti del tipo *aNNForecastAppAgent*, si può notare che il grafico del prezzo presenta una bolla enorme, all'apice della quale il livello del prezzo raggiunge un valore pari a quasi ottocento volte il prezzo iniziale.

La prevedibilità del mercato raggiunge il 58% dei segni corretti nella previsione, il che sembra dimostrare che il mercato presenta una situazione di "profezie autoavveranti".

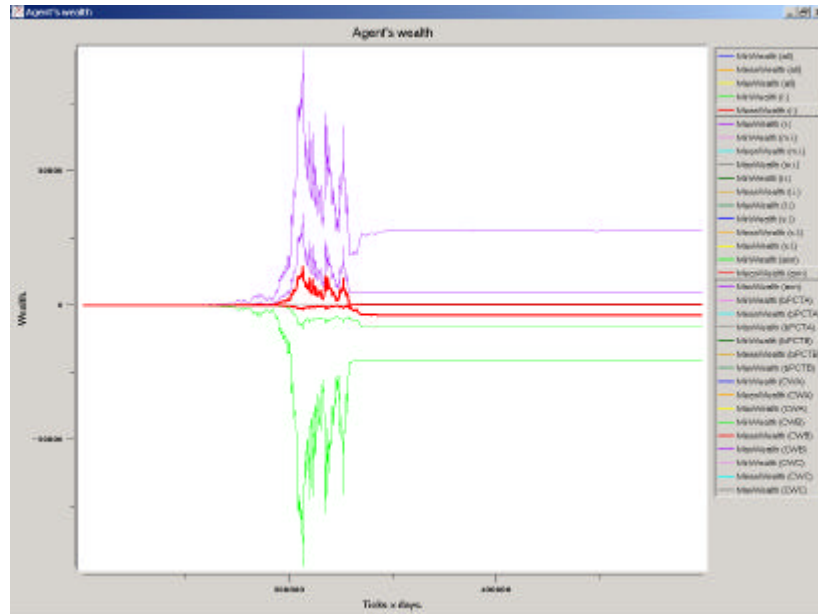


Fig. 5.15 – Grafico della ricchezza con 30 *aNNFoercastAppAgent*, 270 *randomAgent* e *maxOrderQuantity* pari a 3.

### 5.1.8 – Il metodo *CrossTarget*.

Il metodo dei *Cross Target* permette di introdurre all'interno del modello *Sum* degli agenti cognitivi, cioè dotati di "mente", e permette di evidenziare il ruolo di agenti cognitivi all'interno di un mercato strutturato, secondo la struttura evidenziata nel paragrafo 5.1.2.

Questo tipo di agenti utilizza le previsioni effettuate in modo molto simile a quello descritto da Sargent per l'economia reale, gli agenti *Cross Target*, dunque, possono essere considerati, in un certo senso, agenti che applicano aspettative razionali.

L'ipotesi generale sottostante al metodo è che un agente che opera in un ambiente economico deve sviluppare ed adattare la propria capacità di valutare, coerentemente, che cosa deve fare per ottenere un determinato risultato e come calcolare le conseguenze delle proprie azioni. Questo, naturalmente, vale sia per il singolo agente, sia in un ambiente all'interno del quale interagiscono più individui.

Oltre alla coerenza interna, gli agenti devono anche sviluppare altre caratteristiche, ad esempio la capacità di compiere azioni seguendo "Proposte Esterne" (EP), o valutazioni degli effetti seguendo "Obiettivi Esterni" (EO), suggeriti dall'ambiente (ad esempio seguendo delle regole) o da altri agenti.

Queste due caratteristiche aggiuntive sono funzionali per un miglior adeguamento degli agenti durante gli esperimenti.

La definizione *Cross Target* deriva dalla tecnica utilizzata per calcolare i *target* necessari al training delle reti neurali artificiali, che rappresentano gli agenti dell'esperimento.

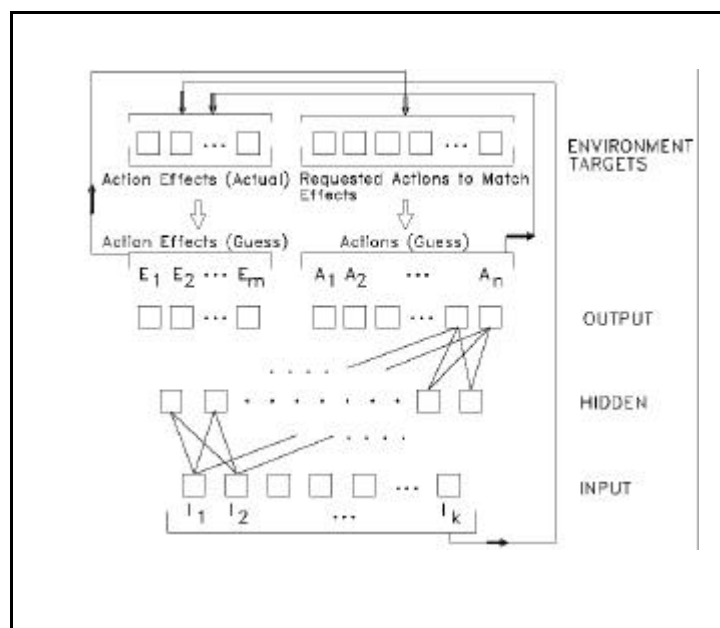


Seguendo l'ipotesi generale, la principale caratteristica di questo tipo di agenti cognitivi, consiste nello sviluppare coerenza interna fra le azioni che compiono e le conseguenze di tali azioni.

Sempre secondo l'ipotesi generale, inoltre, il comportamento degli agenti produce valutazioni che possono essere divise in due parti, l'una che comprende i dati che quantificano le azioni, vale a dire il "che cosa fare", l'altra che comprende le previsioni sulle conseguenze delle azioni.

In questo modo vengono specificati due tipi di output per la rete neurale, allo stesso modo per gli agenti: azioni che devono essere compiute e congetture sugli effetti delle azioni.

La scelta del metodo, che prevede l'uso di una rete neurale, per sviluppare i *Cross Target* è dovuta essenzialmente alla capacità di autoadattamento intrinseca nelle funzioni a rete neurale.



**Fig. 5.16** – Schema della struttura dei Cross Target.

Nella **Fig.5.16** è illustrato lo schema di funzionamento dei *Cross Target*, secondo il quale ogni agente deve produrre delle congetture circa le proprie azioni e gli effetti collegati, sulla base di un set di informazioni che si trovano in input dalla rete neurale.

Secondo il principio di coerenza interna, i target del processo di apprendimento sono, da un lato gli effetti attuali delle azioni del soggetto simulato, dall'altro le azioni necessarie per raggiungere gli effetti congetturati.

Una prima puntualizzazione sul metodo riguarda l'apprendimento, infatti, analizzando i cambiamenti dei pesi durante il processo, si può notare che la matrice dei pesi che legano i nodi di input ed i nodi *hidden* presentano variazioni minime o nulle, mentre la matrice dei pesi che legano il livello *hidden* al livello output variano in modo considerevole.

La conclusione che ne deriva, dunque, è che solo i cambiamenti dei pesi da *hidden* ad output determinano il continuo adattamento delle risposte della rete neurale artificiale ai continui cambiamenti dell'ambiente. Questa situazione è la conseguenza sia delle minime variazioni dei

target, sia del ridotto numero dei cicli di apprendimento, per questo motivo, l'agente economico simulato mostra di sviluppare una capacità locale a prendere decisioni, ma solo in base all'adattamento degli output all'ultimo target, infatti, non vi è alcuna influenza dei valori del livello di input.

Questo tipo di apprendimento viene definito di breve termine, in opposizione all'apprendimento a lungo termine, in particolare si verifica un apprendimento a breve termine quando, nella fase di azione, gli agenti modificano continuamente i propri pesi, soprattutto dal livello *hidden* a quello output, per adattare i target autogenerati con il metodo CT, mentre si ha un apprendimento a lungo termine quando si mappano gli input con i target grazie ad un grande numero di cicli di *learning* e si giunge ad avere una rete neurale artificiale che è definitivamente in grado di applicare le regole implicitamente sviluppate nelle fasi di azione e di apprendimento.

Una seconda puntualizzazione riguarda sia gli obiettivi esterni, sia le proposte esterne, infatti, questi valori possono sostituire i Cross Target nelle fasi di azione e di adattamento e sono coerentemente inclusi nel set di dati utilizzati per l'apprendimento a posteriori.

Considerando i target che derivano dalle azioni, la congettura su un effetto può essere utilizzata per approssimare un valore suggerito da una regola semplice, ad esempio aumentare la ricchezza. Nella terminologia dei *Cross Target* questo è un obiettivo esterno ed il suo effetto indiretto, per mezzo della struttura a *Cross Target*, modificherà le azioni rendendole maggiormente coerenti con le congetture (modificate) sugli effetti.

Al contrario, la congettura su un'azione, per essere compiuta, può essere modificata per mezzo di una proposta esterna, influenzando indirettamente anche le corrispondenti congetture sugli effetti.

Se le proposte esterne, gli obiettivi esterni e il principio di coerenza interna contrastano nel determinare il comportamento, può emergere complessità anche con agenti, sempre in una prospettiva di razionalità limitata, privi di razionalità completa e di un apparato di ottimizzazione.

Riferendosi alla fig.5.15, che illustra il funzionamento secondo uno schema generale dei Cross Target, si può osservare che gli input del modello sono soprattutto i dati provenienti dall'ambiente o dal comportamento di altri agenti, in secondo luogo che essi possono essere indipendenti o dipendenti dalle precedenti azioni del soggetto simulato, infine i target vengono conosciuti solo quando le azioni sono avvenute.

L'algoritmo CT è sia un algoritmo di apprendimento, sia di azione: l'azione è necessaria per produrre le informazioni con le quali vengono costruiti i target per esercitare la rete neurale che simula i soggetti.

Apprendimento ed azioni avvengono in quattro momenti ogni giorno, infatti un giorno è il lasso di tempo determinato dalla somma dei quattro passi necessari per concludere un intero ciclo di stima dell'output, e della propagazione retroattiva dell'errore necessario alla correzione di pesi della rete neurale, che vengono inizializzati in modo casuale in un intervallo.

I quattro passi fondamentali della rete neurale possono essere così schematizzati:

1. Output della rete neurale: le azioni che devono essere compiute, riportate sul lato destro della fig. 5.15 e gli effetti di queste azioni, riportate sul lato sinistro sono congetturate seguendo gli input ed i pesi della rete.
2. Target sul lato sinistro della rete: i target per gli effetti che si suppone derivino dalle azioni, come sono congetturati sul lato sinistro del livello di output, sono calcolati indipendentemente dalle azioni congetturate. In questo modo, le congetture sugli effetti risultano maggiormente legate alle azioni attuali.
3. Target sul lato destro della rete: le differenze calcolate nel passo precedente sui target e sugli output della rete per il lato degli effetti possono essere interpretati inversamente come punti di partenza per la modifica delle azioni, in modo da permettere il match con gli effetti. In questo modo essi vengono usati per costruire i target per il meccanismo che congettura le azioni. (Essendo gli inversi delle formule indicate sotto spesso non definiti, le correzioni sono ripartite a caso fra tutti i *target* da costruire; infine, se diverse correzioni riguardano lo stesso target, viene scelta la maggiore in valore assoluto. In questo modo si vorrebbe imitare il comportamento attuale di un individuo che deve obbedire ad ordini diversi ed incongruenti, probabilmente in questo caso verrebbe seguito l'ordine con il maggior potere imperativo, in questo caso rappresentato dal valore assoluto maggiore.
4. *Backpropagation*: una volta avvenuto l'apprendimento, si correggono i pesi per ottenere che gli effetti congetturati siano più vicini alle conseguenze delle azioni congetturate e le azioni più coerenti con gli effetti congetturati. Dunque si hanno due processi di apprendimento, entrambi basati sulle congetture degli elementi del lato opposto della rete.

Questo doppio processo di adattamento, con l'interazione fra gli agenti ed un apprendimento a lungo termine, assicura l'emergere di comportamenti autosviluppati del tutto non banali, dal punto di vista del percorso dei valori generati dai comportamenti degli agenti.

Gli agenti che, all'interno del modello Sum, utilizzano il metodo dei Cross Target sono di due tipi, i *BPCTAgentA* ed i *BPCTAgentB*, i primi, molto più semplici non vengono quasi mai utilizzati negli esperimenti, in favore dei più sofisticati BPCTB.

Il nome BPCT si riferisce alla *backpropagation* che viene utilizzata per correggere i pesi della rete neurale ed è anche il nome di un pacchetto software scritto anche per *Swarm*, che permette di utilizzare la tecnica dei CT applicandola a diversi tipi di agente.

Gli agenti *BPCTB* hanno otto nodi di input, che sono costituiti dai prezzi medi dal giorno  $t - 5$  al giorno  $t - 1$  (al giorno  $t$ ) e la stima dell'indice di prezzo per il giorno  $t + n - 1$ , in cui  $n$  è uguale al parametro *nAheadForecasting*, che determina a quanti giorni avanti si riferisce la previsione.

I nodi *hidden* sono sei e, secondo lo schema Cross Target, ci sono quattro nodi di output per il lato degli effetti e quattro dal lato delle azioni.

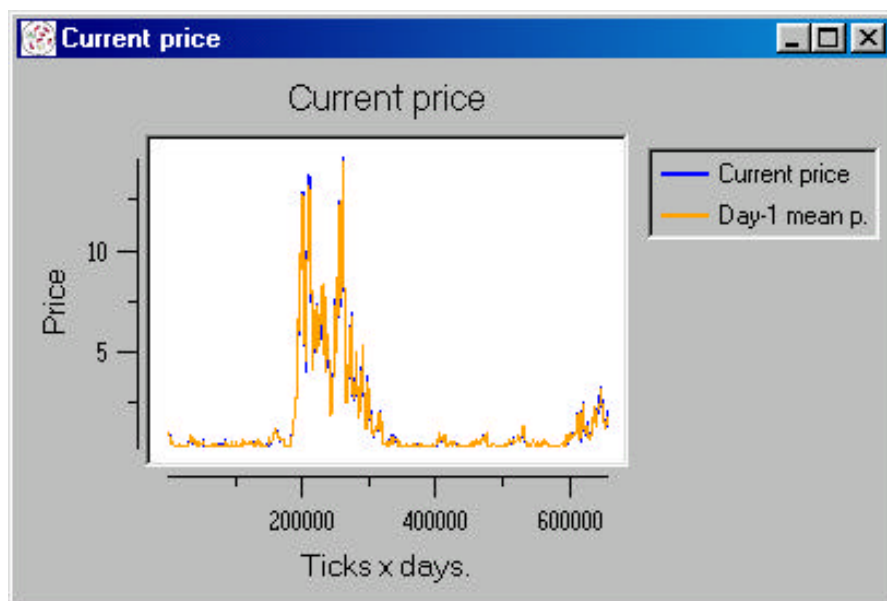
Gli effetti, calcolati alla fine del giorno  $t$ , sono la liquidità, la quantità di azioni, la ricchezza calcolata in base all'ultimo prezzo di chiusura e la ricchezza calcolata in base al prezzo previsto.

La scelta di acquistare o vendere è rappresentata, secondo una metrica esterna, da un numero reale entro l'intervallo compreso fra  $-maxOrderQuantity$  e  $+maxOrderQuantity$ ; se tale valore,  $bs$ , è positivo, l'agente vende, se è negativo, l'agente acquista, se è pari a zero, cosa peraltro decisamente rara, l'agente non opera; la quantità acquistata o venduta, o il numero di ordini unitari, è determinata arrotondando  $|bs|$  all'unità superiore.

Gli obiettivi esterni sono legati ai due effetti sulla ricchezza e sono determinati aggiungendo direttamente alle congetture interne sulla ricchezza una quantità fissa,  $d$ .

Gli obiettivi esterni sono di tre tipi diversi, e corrispondono: ad un obiettivo di incrementare la ricchezza calcolata sul prezzo di chiusura (EO=1), ad un obiettivo di incrementare la ricchezza calcolata con il prezzo della previsione (EO=2), ed al tentativo di incrementare entrambe contemporaneamente (EO=3).

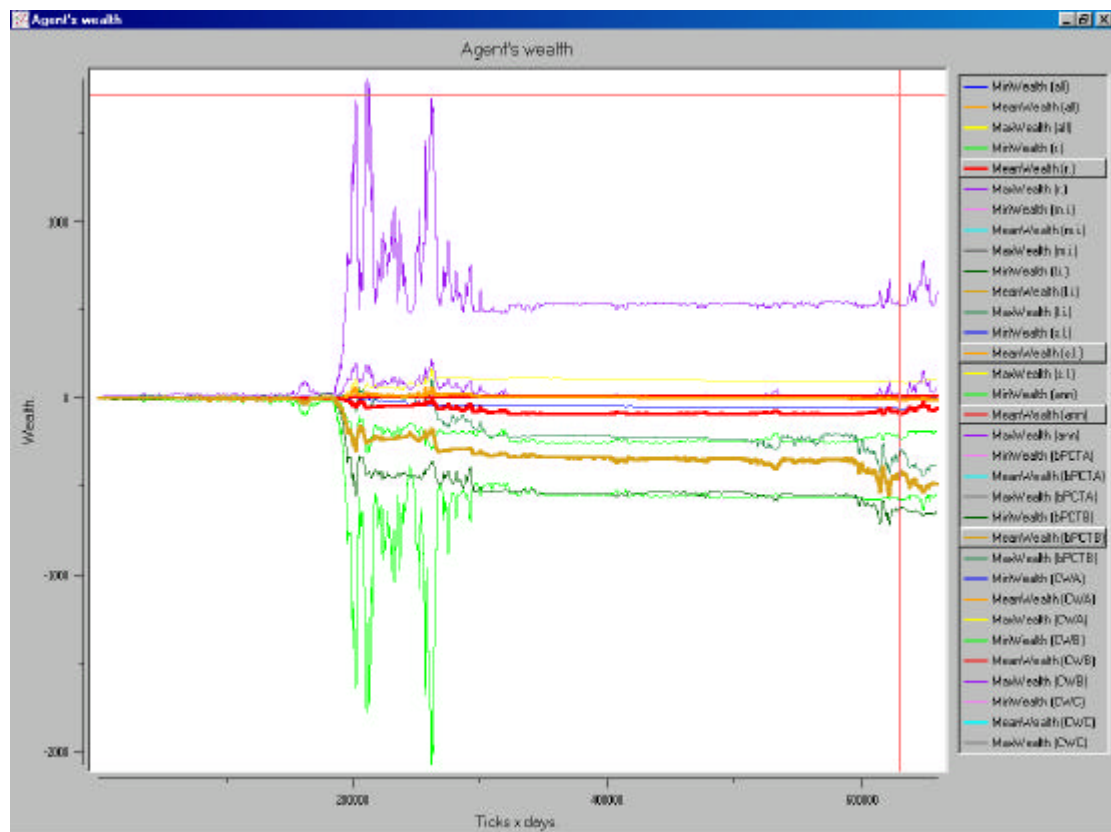
Gli esperimenti condotti con agenti cognitivi BPCTB prendono in considerazione simulazioni con diversi valori per il parametro  $d$  per rendere più o meno importanti gli obiettivi esterni di tipo 2 ed utilizzano anche, per motivi essenzialmente di confronto, obiettivi esterni di tipo 1.



**Fig.5.17** – Grafico del prezzo Agenti CT: 270 random, 15 ann, 10 stop loss, moq=3, 5 BPCTB, EO=2, d=10.

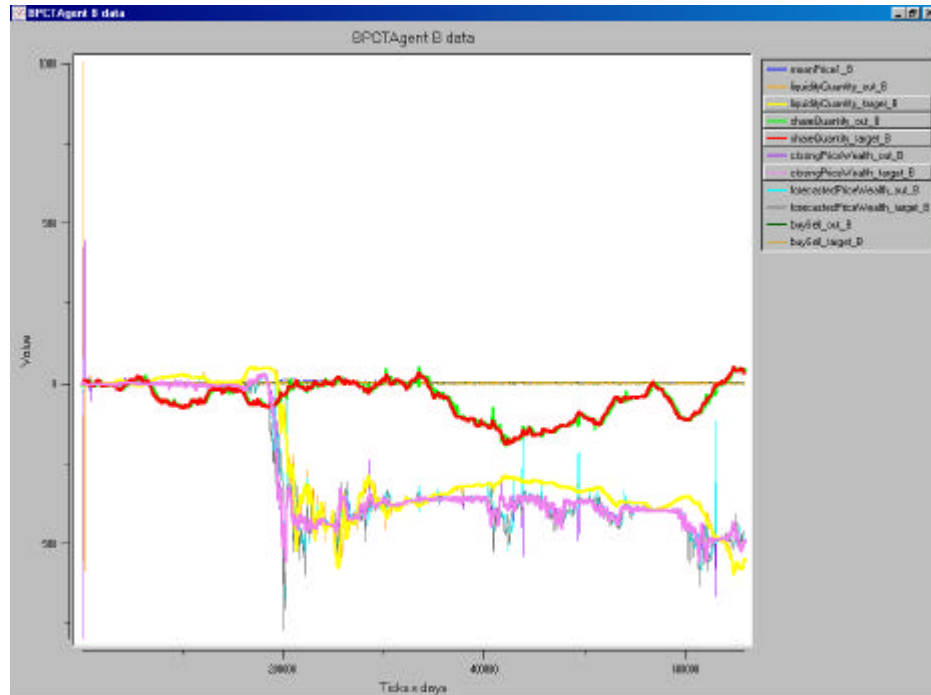
Il primo caso considerato riguarda un esperimento con agenti del tipo BPCT, che hanno l'obiettivo di aumentare la ricchezza sulla base del prezzo previsto (EO = 2) e  $d = 5$ .

Il grafico presenta una sola bolla abbastanza rilevante e il livello di prevedibilità del mercato è del 54%, conferma cioè la prevedibilità di un mercato in cui convivono agenti di diverso tipo, *random*, *ANNAppIForecasting* e *stop loss*, con solo l'aggiunta dei BPCT.



**Fig. 5.18** – Grafico del livello di ricchezza con 270 agenti random, 15 aNNForecastAppl, 10 stop loss, moq=3, 5 BPCTB, EO=2, d=10.

Analizzando i risultati del primo agente CT, si può notare l'obiettivo esterno, calcolato con  $d=5$ , non esercita alcun effetti sulla ricchezza dell'agente, valutata sulla base del prezzo previsto. In corrispondenza della bolla, il livello di ricchezza diventa negativo e anche una notevole varianza della quantità di azioni non cambia il risultato, il parametro  $d$  con questo valore, dunque, rende l'obiettivo esterno troppo debole e quasi del tutto influente sul comportamento dell'agente CT.



**Fig. 5.19** - Dati di un agente CT.

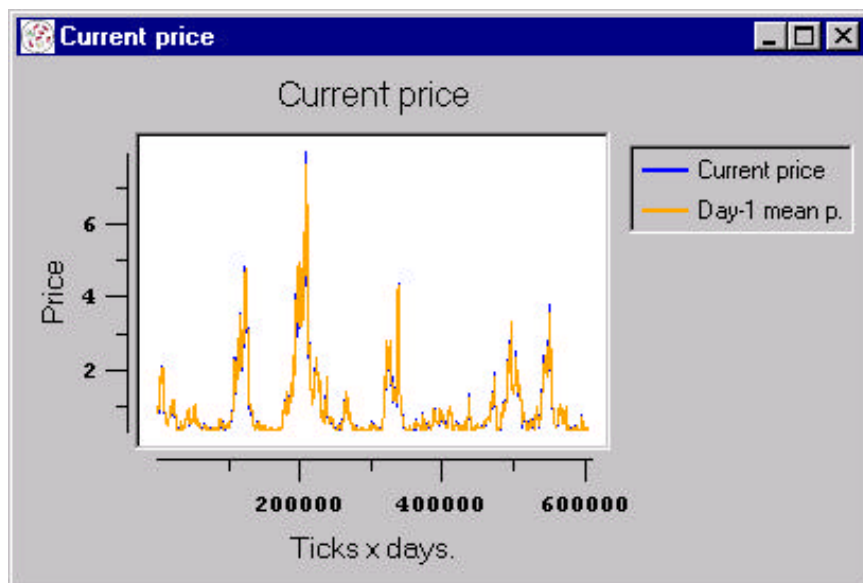
Nel secondo caso esaminato si mantiene lo stesso obiettivo esterno di tipo 2, cioè aumentare la ricchezza calcolata sulla base del prezzo previsto, ma lo si rafforza aumentando il parametro  $d$  a 10.

In questo caso, durante i 2000 giorni della simulazione, si verifica una sequenza di prezzi realistica, che presenta due bolle durante le quali il prezzo raggiunge al massimo 10 volte il prezzo iniziale. Il mercato presenta un livello di prevedibilità pari al 56%.

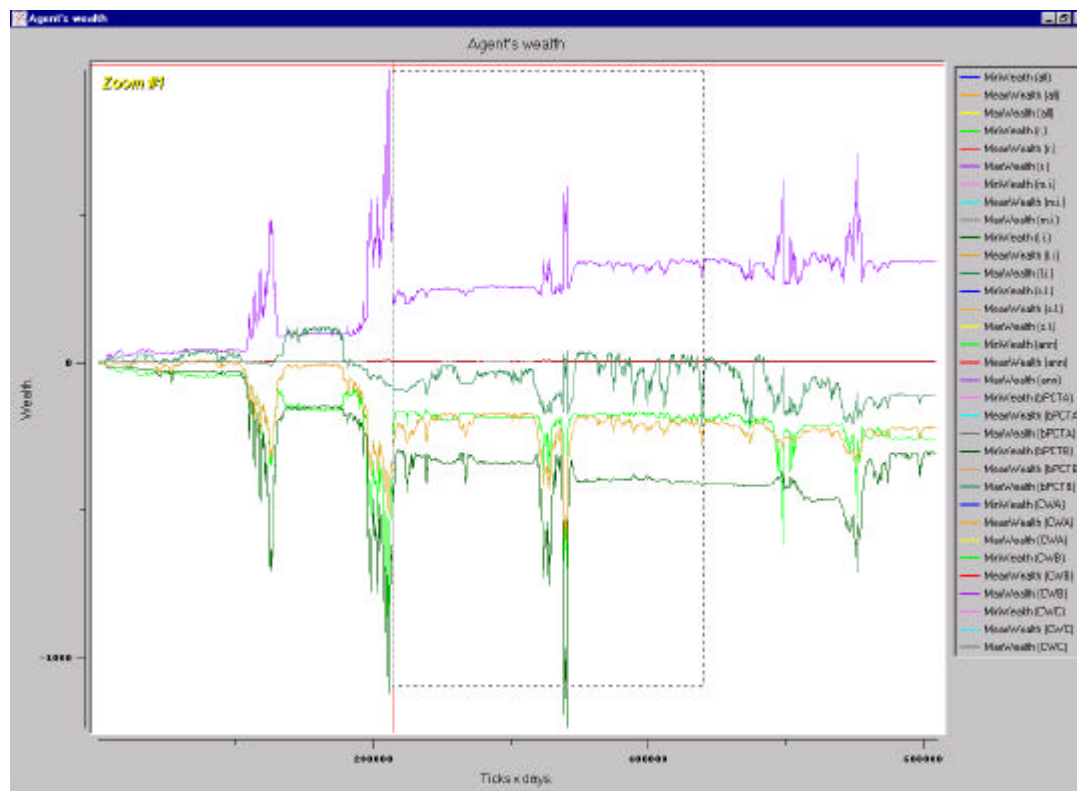
La caratteristica interessante che si può notare, guardando il grafico che rappresenta il livello di ricchezza degli agenti, è costituita dal fatto che durante le due bolle, gli agenti CT riescono ad incrementare, anche in modo significativo, il proprio livello di ricchezza, anche se in modo non del tutto stabile.

La ricchezza viene incrementata grazie alla gestione corretta della quantità di azioni, ma non si può dimenticare che questo tipo di comportamento è autosviluppato dagli agenti solo grazie al metodo dei *CrossTarget* e ad un apprendimento di lungo termine.

Un obiettivo esterno di media intensità, dunque, sembra consentire agli agenti di sviluppare le regole corrette per reagire ai cambiamenti ambientali con sufficiente efficacia.



**Fig. 5.20**– Grafico del prezzo in presenza di 270 agenti random, 15 agenti *aNNApp...*, 10 stop loss e 5 BPCTB con EO 1 e  $d=10$



**Fig. 5.21** – Grafico della ricchezza in presenza di 270 agenti random, 15 agenti *aNNForecastApp*, 10 stop loss e 5 BPCTB con EO 1 e  $d=10$

Prendendo in considerazione un mercato, in cui operano agenti del tipo BPCTB con l'obiettivo esterno di aumentare il proprio livello di ricchezza sulla base della valutazione al prezzo effettivo di chiusura, si può notare che tale mercato presenta sì una sequenza di prezzi abbastanza realistica, caratterizzata dalla presenza di una sola bolla, ma che la prevedibilità è scesa al 49%.

Gli agenti di questo tipo non sono in grado di sviluppare una strategia efficace, anche in assenza di bolle.

Questa prima parte di simulazioni dimostra in primo luogo che l'obiettivo di creare mondi artificiali con agenti semplici alla Axelrod, è perseguibile e offre risultati decisamente soddisfacenti, infatti, come si è potuto vedere anche dall'interazione di agenti completamente casuali emergono fenomeni complessi, nel caso specifico emergono bolle e crash nella sequenza dei prezzi di un mercato di borsa.

In secondo luogo si può notare che le strutture esterne del mercato hanno una capacità del tutto simile ai modelli razionali di generare emergenza di fenomeni difficilmente spiegabile ed infine, che sulla base di questo modello, dall'interazione degli agenti all'interno del mercato, possono emergere comportamenti "intelligenti" autosviluppati con il procedere della simulazione.



## **BIBLIOGRAFIA CAPITOLO QUINTO.**

- AXTELL R. (2000), *"Why Agents? O the varied motivations for agent computing in the Social Science."* In Conte R., Hegselmann R., Terna P. (eds.). *Simulating Social Phenomena, Lecture Notes in Economics and Mathematical Systems 456*, pp.21-40, Berlin, Springer.
- BELTRATTI A., MARGARITA S., TERNA P. (1996), *Neural Networks for Economics and Financial Modelling*. London, International Thompson Computer Press.
- SARGENT T.J. (1993), *Bounded Rationality in Macroeconomics*. Oxford, Clarendon Press.
- TERNA P. (2000), *The "mind or no mind" dilemma in agents behaving in a market*, in G. Ballot e G. Weisbuch (eds.), *Applications of Simulation to Social Sciences*. Paris, Hermes Science Publications.
- TERNA P. (2001), *Cognitive agents behaviors in a simple stock market structure*. In corso di pubblicazione.

## **Capitolo Sesto: il modello SUM con l'uso di *classifier system*.**

L'introduzione, nel modello SUM, di agenti cognitivi che utilizzano *classifier system* per sviluppare le proprie regole di comportamento nel mercato, ha in primo luogo lo scopo di valutare gli effetti dell'introduzione di agenti, che autosviluppano le proprie regole, sulla ricchezza degli agenti, sulla stabilità del mercato e sulla sua prevedibilità, e, in secondo luogo, quello di valutare le analogie o le differenze rispetto al comportamento di agenti cognitivi che utilizzano il meccanismo dei *Cross-Target*.

La diversa struttura di questi due tipi di agenti, infatti, giustificherebbe comportamenti anche molto diversi, dal momento che gli agenti costruiti il metodo dei *Cross Target* vengono, in un certo senso, spinti verso un determinato comportamento, grazie al principio della coerenza interna ed al raggiungimento di obiettivi esterni, mentre gli agenti costruiti con il metodo dei *classifier system* dovrebbero imparare a reagire agli stimoli ambientali solo sulla base dell'esperienza passata e della remunerazione ottenuta in base ai risultati conseguiti.

Per costruire gli agenti che utilizzano *classifier system*, è stato utilizzato un pacchetto software, che permette di applicare questa tecnica evolutiva a diversi tipi di rappresentazioni economiche, e che è stato sviluppato per una tesi di laurea della Facoltà di Economia dell'Università degli Studi di Torino, di cui si può leggere in maniera molto più approfondita in Ferraris (1999).

### **6.1.1 – Il pacchetto CW (*Classifier Workbench*).**

Il pacchetto *Classifier Workbench (CW)* consiste in un modello *Swarm* completo, scritto per permettere di applicare l'algoritmo dei *classifier system* a simulazioni sociali ed economiche dalla struttura diversa.

CW è basato sul paradigma mentale degli algoritmi evolutivi: i singoli cromosomi, che in questo caso sono costituiti da regole, sono considerati come idee nella "mente" dell'agente. Il *RuleMaster* provvede a selezionare, di volta in volta, l'idea migliore da applicare, che dipende dalla descrizione delle condizioni ambientali fatta dall'agente, e suggerisce all'agente stesso quale azione effettuare nell'ambiente fra quelle possibili.

Dopo aver operato, l'agente deve comunicare al *RuleMaster* la valutazione della regola che ha applicato, tale valutazione influisce direttamente sulle possibilità di sopravvivenza e di riproduzione della regola che ha suggerito l'azione.

Per evolvere la popolazione di idee il *RuleMaster* si avvale dei servizi del *RuleMaker*, che è in grado di effettuare la manipolazione genetica delle singole idee, attraverso l'applicazione di un algoritmo genetico.

E' possibile utilizzare CW nell'ambito di una concezione individuale, in cui ogni cromosoma rappresenta un intero individuo, assegnando ad ogni agente un magazzino di regole a cui attingere, assegnando cioè un *DataWarehouse* per ogni singolo agente, oppure far condividere un unico *DataWarehouse* da tutti gli agenti, in modo tale da avere un solo cromosoma.

Il nucleo di CW è un *learning-classifier-system*, cioè un sistema di inferenza basato su regole che evolvono, modificandosi sulla base della valutazione che ogni singolo agente effettua sull'azione che gli è stata suggerita dal *RuleMaster*.

In base alle valutazioni degli agenti, ogni regola accumula un patrimonio di crediti che permette di quantificarne l'efficacia all'interno dell'ambiente, l'apprendimento, o meglio, l'evoluzione delle regole avviene così, non solo in base ai metodi classici degli algoritmi genetici, cioè la mutazione e l'incrocio delle regole esistenti, ma anche tramite una selezione delle regole che si sono rivelate più efficaci e, quindi, hanno un livello di *fitness* più elevato.

Le azioni di mutazione ed incrocio possono essere variate o annullate, semplicemente modificando dei parametri, e, quindi, CW può essere configurato, senza ulteriori modifiche del *software*, sia come *learning-classifier*, sia come *non-learning-classifier*, in grado di effettuare soltanto la selezione delle regole migliori da una popolazione iniziale ed in grado di comportarsi da sistema esperto tradizionale, capace di applicare regole predeterminate in risposta a situazioni diverse; questo vale, sia nel caso di utilizzo secondo il paradigma individuale, in cui ciascun agente possiede un proprio patrimonio di regole, sia nel caso di un paradigma collettivo, in cui il *DataWarehouse* è condiviso da tutti gli agenti, solo che nel primo caso la selezione, anziché essere sulle regole, viene fatta sugli agenti stessi.

La classe di oggetti *DataWarehouse* del pacchetto CW possiede i metodi per creare, automaticamente e in modo casuale, una popolazione iniziale di regole o, ugualmente, per caricarle interamente da un file esterno, questo fa sì che sia possibile far interagire entità che si trovano a

stadi di evoluzione diversi e che sono dotate di insiemi di idee differentemente numerosi, concepiti per codificare condizioni ed azioni proprie di ciascun agente o comuni a più agenti.

In conclusione CW, utilizzando un unico *RuleMaster* ed un unico *RuleMaker*, può gestire parallelamente le decisioni relative a più agenti, variamente adattivi, secondo tre tipi fondamentali: agenti che operano seguendo un *learning classifier*, agenti esclusivamente capaci di selezionare il comportamento migliore fra una serie di possibilità, *non-learning-classifier*, agenti puramente reattivi, che operano in base ad un sistema esperto rigido, capace soltanto di associare una predeterminata azione ad una condizione contingente.

Ciascuno di questi tre tipi di agenti, va visto come un sistema autonomo, che deriva dall'adozione di parametri diversi l'erogazione dei servizi ai diversi oggetti, da parte dell'unica coppia di oggetti di servizio, *RuleMaster* e *RuleMaker*, in base a quanto specificato nel *dataWarehouse* di ciascun agente.

Inoltre, dotando ogni agente di un diverso gestore di parametri, *classifierParm*, è possibile differenziare ulteriormente il comportamento dei singoli, all'interno delle tre tipologie, spingendosi al livello limite di dotare ciascun agente di un proprio differente sistema decisivo.

Aderendo all'impostazione mentale, ogni agente deve disporre di un proprio patrimonio di regole, soggetto ad un'evoluzione autonoma e, secondo l'impostazione dettata dallo schema ERA in Terna (1998), non possono comunicare direttamente con gli altri agenti, nel *DataWarehouse* di ogni agente sono contenute tutte le informazioni relative al suo stato evolutivo, rappresentati dagli indirizzi dei gruppi di oggetti che formano il suo genotipo, dall'indirizzo della regola attiva, cioè della regola che ha proposto l'ultima azione, dall'indirizzo dell'oggetto contenente i parametri del *classifier* per se stesso, dai contatori delle varie azioni evolutive effettuate, dall'eventuale nome del *file* da cui effettuare il *load* iniziale della popolazione di regole.

Ogni agente conosce anche l'indirizzo di un' *Interface* specifica o condivisa con altri agenti che trattano lo stesso tipo di problemi, la condivisione dell'*interface*, infatti, non è possibile se gli agenti devono trattare problemi diversi, poiché le metriche interne risultano differenti. In questo caso sarebbe meglio diversificare anche i *classifierParm*, anche se la classe possiede dei metodi che permettono di variare dinamicamente il valore dei parametri.

Infine l'agente deve conoscere l'indirizzo del *RuleMaster*, che può anche essere unico per tutto il modello, e deve anche essere in grado di recepire informazioni dall'ambiente per comunicarle al *RuleMaster* tramite l'interface. Allo stesso modo il *RuleMaster* comunica agli agenti i suoi suggerimenti. In questo modo il *classifier* ha una visione dell'ambiente che risulta mediata dalle percezioni dell'agente che ne richiede i servizi, facendo sì che il comportamento degli agenti risulti più vario e riesce a cogliere con maggior precisione l'eterogeneità ravvisabile nel mondo reale.

Il pacchetto CW è in grado di gestire più *classifier* contemporaneamente, derivanti da diverse combinazioni di *DataWarehouse* e *ClassifierParm*, infatti il nucleo di CW è in grado di effettuare le operazioni di attribuzione dei crediti e di evoluzione della popolazione di regole, in base ad un algoritmo genetico, agendo su insiemi diversi di regole e messaggi ed utilizzando configurazioni

differenti di parametri. Ogni combinazione di parametri ed insiemi di regole caratterizza un diverso *classifier*, anche se gli oggetti incaricati delle operazioni restano sempre gli stessi.

Gli insiemi di regole e messaggi sono rappresentati da gruppi di oggetti, organizzati in base ai protocolli delle liste di Swarm, attraverso le quali è possibile accedere a tutti gli elementi che le compongono contemporaneamente, senza che sia necessario memorizzare tutti gli indirizzi dei singoli oggetti.

L'intero sistema è basato sull'interazione delle istanze di quattro classi: *Rule*, *Message*, *Effector* e *Treasury*. Il *dataWarehouse* di ciascun agente contiene una o più istanze di queste classi, solitamente raggruppate in liste, le liste utilizzate sono sei: *ruleList*, *messageList* ed *effectorList* contengono gli indirizzi degli oggetti nominati, la *matchList* è utilizzata per le regole che soddisfano la condizione di *match* con le informazioni ambientali fornite dall'agente, la *winnerList* è utilizzata per le regole che hanno vinto l'asta, e la *workList*, che è una lista di appoggio per l'elaborazione.

Per quanto riguarda le regole, invece, ciascuna regola è costituita da un cromosoma formato da due genomi, che rappresentano rispettivamente la *condition-part*, o < condizione > e la *action-part*, o < azione >, l'estensione dei due genomi, cioè il numero di posizioni utilizzate per la codifica, può variare fino ad un massimo stabilito, facilmente modificabile.

Ogni posizione, che corrisponde alla presenza o assenza di un carattere ambientale, può assumere i valori {0,1,#}, quest'ultimo simbolo è un carattere che rappresenta l'indifferenza rispetto alla presenza o assenza della caratteristica osservata, è una *wildcard*. Iti di ciascuna regola

Apposite variabili contengono l'ammontare del patrimonio di crediti di ciascuna regola, *strenght*, il valore di specificità della regola stessa, *specificity*, e l'ammontare dell'offerta, fatta da ciascuna regola per partecipare all'asta in base alla quale si determina quali individui avranno diritto di immettere il proprio messaggio nella *messageList*.

Ogni regola, inoltre, possiede una *partnerList*, in cui memorizza gli indirizzi delle entità autrici dei messaggi che hanno permesso l'attivazione della regola stessa come conseguenza del *match*.

I messaggi possiedono un cromosoma costituito da un solo genoma, contenente il testo del messaggio, cioè la traduzione delle informazioni ambientali o la copia della *action-part* di una regola; inoltre, in ciascun messaggio è indicato l'indirizzo del suo autore, *owner*.

Gli *effector* sono simili alle regole, ma la loro *action-part* contiene la codifica, binaria, di un numero che indica l'azione da consigliare agli agenti, che viene tradotta nella metrica dell'agente dall'interface. Anche gli *effector*, come le regole, possiedono una *partnerList*, in cui memorizzano l'indirizzo delle regole che li hanno attivati; un'apposita variabile è utilizzata per registrare il valore di *support*, cioè la somma delle offerte fatte da tutte le regole che hanno attivato l'*effector*, utile per decidere quale, fra tutti gli *effector* attivi, debba prevalere e suggerire all'agente l'azione di cui è portatore.

```
479A378 strength: 0.863353 specificity  
1.000000  
1011 → condition-part  
0011 → action-part  
My partnerList contains 1 entries for:  
4795878m
```

**Fig.6.1-** Esempio di Regola

```
4795CD8 suggestion 3  
0011 → action-part  
My partnerList contains 1 entries for:  
479A378
```

**Fig.6.2** – Esempio di *Effector*

La *treasury* è la controparte dei movimenti di crediti effettuati durante l'elaborazione, fornisce il numerario per pagare le ricompense, incassa le *bid taxes* e gestisce le ricompense.

Quando viene richiesto un consiglio al *ruleMaster*, questo provvede a caricare i dati dal *dataWarehouse* indicato dall'agente ed in particolare provvede a caricare gli indirizzi delle sei liste presenti nel *dataWarehouse*, l'indirizzo del file che gestisce i parametri del *classifier* e, interrogando quest'ultimo, i valori dei parametri di funzionamento.

Subito dopo il *ruleMaster* interroga l'*interface*, il cui indirizzo è il valore del secondo parametro ricevuto dall'agente, per conoscere i dati relativi all'ambiente, con questi dati il *ruleMaster* costruisce un messaggio e lo immette nella *messageList* dopo quelli provenienti dalle regole, residuati dal ciclo precedente.

Il terzo passo è costituito dalla ricerca delle regole che soddisfano il *match* con almeno uno dei messaggi della lista, le regole che soddisfano il *match* vengono inserite nella *matchList* precedentemente svuotata.

Se nessuna regola soddisfa il *match*, cioè se la *matchList*, dopo il confronto di tutte le regole con tutti i messaggi, risulta vuota, un componente specifico, il *cover detector*, provvede alla modifica della regola più vicina al *match* e con il patrimonio di minor ammontare, così da garantire l'avviamento del meccanismo, infatti, in questo modo viene garantita l'attivazione di almeno una regola, ma con l'immissione di un turbamento minimo nel sistema, sia perché si modificano poche posizioni della regola, sia perché l'individuo modificato, avendo un patrimonio poco consistente, probabilmente è poco importante. Le liste di regole e messaggi vengono mescolate casualmente, per mezzo dell'oggetto *Shuffler*, per immettere movimento nel sistema.

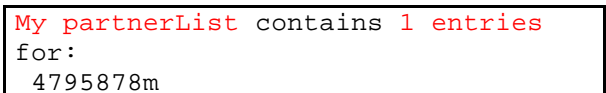
Il quarto passo consiste nella conduzione dell'asta: a tutte le regole presenti nella *matchList* viene richiesto di effettuare un'offerta, *bid*, in base ai parametri del *classifier*, in seguito si procede

a selezionare la regola che ha effettuato l'offerta maggiore e, una volta individuata, questa viene spostata dalla *matchList* ed immessa nella *winnerList*, precedentemente svuotata.

La procedura viene ripetuta fino a selezionare un numero di vincitori pari alla quantità massima specificata nell'oggetto di gestione dei parametri. Per ridurre la possibilità di avere bid di eguale ammontare, ogni regola viene selezionata in base ad un valore, *effective bid*, ottenuto sommando al bid stesso un valore casuale, compreso fra zero ed uno, che viene mantenuto sufficientemente piccolo grazie alla moltiplicazione per un ulteriore valore specificato nei parametri.

A questo punto le regole vengono tassate di una quota proporzionale al loro patrimonio, con due tipi di imposta, una che riguarda tutte le regole, *life tax*, ed un'altra che, invece, riguarda solo le regole che partecipano all'asta, *bid tax*, le aliquote di questi due tipi di imposta sono determinate dall'oggetto di gestione dei parametri e vengono, invece, incassate dalla *treasury*.

Il sesto passo consiste nello svuotare la *messageList*, eliminando tutti i messaggi in essa contenuti, poiché sono oramai divenuti inutili, dal momento che il loro testo ha già partecipato ad una sessione di *matching* e che le regole attivate hanno già registrato, nella loro *partnerList*, l'indirizzo degli autori dei messaggi.



```
My partnerList contains 1 entries
for:
4795878m
```

**Fig. 6.3** – Indicazione del possibile match, fatta da una regola.

A questo punto, tutte le regole presenti nella *winnerList* ricevono l'ordine di pagare i rispettivi *bid* agli autori dei messaggi che ne hanno consentito l'attivazione, cioè alle regole presenti nella loro *partnerList*, e contestualmente ricevono l'ordine di emettere i propri messaggi. Ciascuna regola comunica il contenuto della propria *action-part*, che viene immesso in un messaggio da aggiungere alla *messageList*.

In base al contenuto della nuova *messageList*, viene ricercato il match con gli *effector*: ciascuno di quelli che soddisfano il match provvede a registrare l'indirizzo della regola, proprietaria del messaggio, nella propria *partnerList* ed a sommare, al proprio valore di *support*, l'ammontare del *bid* della regola. Gli *effector* attivati dal match vengono a loro volta inseriti nella *matchList*, precedentemente svuotata. Analogamente a quanto avviene per l'inserimento delle regole nella *matchList*, se tale lista dovesse risultare vuota al termine del processo, verrebbe applicata un'apposita *routine*, *cover effector*, volta a garantire almeno un *match*. Infine, tutti i messaggi utilizzati dagli *effector* vengono eliminati dalla lista in modo tale che non possano attivare altre regole.

L'ultimo passo del procedimento consiste nell'effettuare la scelta dell'*effector* che avrà diritto di consigliare l'azione in base al proprio valore di *support*. L'indirizzo di questo *effector* viene registrato nel *dataWarehouse* e viene richiamata l'*interface* affinché registri il valore dell'azione, espresso nella metrica interna, dall'*action-part* dell'*effector* vincitore.

Un aspetto fondamentale del *classifier system*, oltre alla gestione vera e propria del *classifier*, che è stata analizzata nel dettaglio, è quello dell'evoluzione delle regole.

Quest'attività, nel pacchetto CW, è affidata ad un oggetto della classe *Rulemaker*, che è in grado di trattare diverse popolazioni contemporaneamente, ricostruendo di volta in volta, lo specifico algoritmo genetico da applicare, in base ai parametri utilizzati da ciascun tipo di agente.

Il *ruleMaker* interagisce esclusivamente con i *dataWarehouse*, leggendo e modificando i dati in essi contenuti, l'attivazione del programma è operata dal *ruleMaster*, che provvede a comunicare al *ruleMaker* l'indirizzo del *dataWarehouse* su cui operare.

Dal *dataWarehouse*, il programma ottiene gli indirizzi della lista delle regole e dell'oggetto di gestione dei parametri per lo specifico *classifier*. Gli individui "genitori" sono selezionati a coppie e da ciascuna coppia vengono generati due nuovi individui, che vanno a sostituire i due peggiori e, contestualmente, più simili ai nuovi nati, che vengono scelti da un campione della popolazione.

Questa procedura viene ripetuta fino a raggiungere la quota di ricambio generazionale specificata nei parametri, che può essere personalizzata in modo diverso per ciascun agente.

Il primo passo per l'evoluzione delle regole, consiste nell'acquisizione delle notizie relative alla popolazione su cui operare, in base alle quali viene determinato il numero di riproduzioni da effettuare e la ricchezza totale di tutte le regole, successivamente si avvia la procedura di riproduzione per ciascuna coppia selezionata; al termine di ogni riproduzione, i nuovi nati andranno a rimpiazzare due individui della popolazione corrente.

Gli individui "genitori" vengono selezionati in base al proprio patrimonio in modo stocastico, cioè si procede sommando i patrimoni delle regole fino ad ottenere un ammontare superiore ad un valore *random*, compreso fra zero e la ricchezza totale e viene scelta l'ultima regola che ha partecipato alla somma. In questo modo, ciascuna regola riceve una probabilità di riprodursi proporzionale al suo patrimonio, cioè, più la regola è ricca e possiede una *fitness* elevata, più è alta la possibilità che l'addizione del suo valore comporti il superamento della soglia, casualmente stabilita.

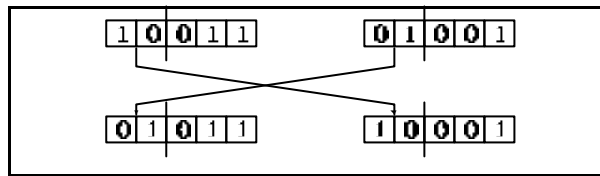
La generazione pseudo-casuale del valore di soglia avviene in base ad una distribuzione di probabilità uniforme. Il procedimento viene ripetuto per la scelta del secondo genitore, avendo cura di evitare che la scelta ricada sulla regola già individuata come primo genitore.

I patrimoni genetici dei due genitori, a questo punto, vengono incrociati ed assoggettati a mutazione, in base ai parametri validi per lo specifico *dataWarehouse*. L'attività di *crossing-over* viene condotta in modo da mantenere la distinzione fra *action-part* e *condition-part*, incrociando separatamente le due parti delle regole.

Il crossover viene effettuato generando in modo casuale due interi, uno per la condizione ed uno per l'azione, di valore compreso fra uno e la lunghezza di ciascuna parte, diminuita di un'unità

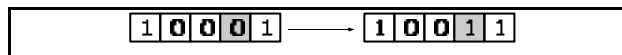
In base a questo valore i due genomi possono essere divisi, a loro volta, in due parti, sinistra e destra: ciascun figlio riceve un cromosoma ottenuto copiando quello dei genitori e scambiando le parti sinistre dei due genomi.





**Fig. 6.4** – Crossover

La copia viene effettuata per singole posizioni , prima di ogni copia si sottopone a mutazione l'allele, secondo la probabilità specificata nei parametri impostati. La mutazione consiste nel cambiare il valore di una singola posizione della stringa genetica: da zero ad uno, da zero a *wildcard* (#), da uno a *wildcard*, da uno a zero, da *wildcard* a zero e, infine, da *wildcard* ad uno.



**Fig. 6.5** - Mutazione

Può accadere che nuovi individui risultino uguali ad altri già esistenti, e, in questo caso, l'aggiunta di una regola non comporterebbe nessuna novità effettiva, per questo motivo i nuovi cromosomi vengono sottoposti a verificaione , cioè vengono confrontati con quelli già esistenti per verificare che non vi siano copie, in caso di esito negativo, si riprende il procedimento a partire dalla selezione dei genitori, fino ad ottenere individui dotati di originalità rispetto a quelli esistenti.

Solo a questo punto è possibile creare due nuove regole, costituite dai nuovi cromosomi e dotate di un patrimonio iniziale pari alla media dei patrimoni dei genitori.

L'ultima fase del procedimento di evoluzione delle regole consiste nella selezione dei due individui da rimpiazzare. Il criterio scelto per questo tipo di operazione prevede che venga sostituita la regola che sia , contemporaneamente, dotata di minor patrimonio e che sia maggiormente simile a quella destinata a prendere il suo posto, in modo tale da non immettere nell'insieme delle regole un'eccessiva turbativa.

La selezione viene condotta su una porzione di popolazione che si ottiene estraendo dalla popolazione totale un campione determinato dai parametri *crowdingFactor* e *crowdingRate*, moltiplicando il numero delle regole per il parametro *crowdingFactor*, si ottiene la numerosità di un primo campione e ciascun individuo appartenente a questo campione viene confrontato con tutti gli individui appartenenti ad un secondo campione ottenuto moltiplicando lo stesso numero di regole per il parametro *crowdingRate*.

Ad ogni confronto viene scelta la regola dotata di minore patrimonio e, successivamente le regole selezionate vengono confrontate con il nuovo individuo da immettere nella popolazione, in modo tale da individuare la regola più simile a quest'ultimo, per eliminarla dalla *ruleList* e distruggerla. Dato che la nuova regola, poi, viene accodata alla lista, occorre ricalcolare il valore della somma dei patrimoni.

Un'ulteriore chiarimento , per quel che riguarda il pacchetto CW, riguarda la traduzione, a carico dell'*interface*, dalla metrica esterna, o dell'agente, a quella interna, o del *classifier*.

Utilizzando metodi genetici, questo aspetto è particolarmente rilevante, perché sia gli algoritmi genetici, sia i *classifier system*, operano sulla rappresentazione dei dati, prescindendo dal significato semantico attribuito dall'agente che, secondo l'impostazione dettata dalla struttura ERA, è conosciuto solo dall'*interface*.

In questo modo, l'agente opera in base alla propria percezione degli eventi ambientali, indicandoli con i costrutti grammaticali scelti dal programmatore, ed esegue azioni codificate in modo analogo, lasciando all'*interface* il compito di tradurre queste informazioni nelle sequenze di simboli che il *classifier* è in grado di accettare, cioè le stringhe binarie definite dall'uso dei soli simboli {0, 1, #} e, successivamente, di tradurre nuovamente i suggerimenti del *classifier* nella metrica dell'agente.

### **6.1.2 – Applicazione dei classifier system al modello SUM.**

L'applicazione dei *classifier system*, in particolare del pacchetto CW, al modello SUM serve per introdurre agenti cognitivi, che sono in grado di elaborare le proprie strategie di azione non sulla base di un apprendimento a rete neurale, ma sulla base di un algoritmo evolutivo.

La struttura del *classifier system*, illustrata in modo approfondito nel paragrafo precedente, è stata applicata ad agenti che operano all'interno del modello Sum, per valutare quali effetti essi possono avere sull'andamento del prezzo di mercato, sulla stabilità e sulla prevedibilità del mercato stesso.

Il secondo obiettivo dell'introduzione di questo tipo di agenti consiste nel confrontare le regole di comportamento sviluppate con quelle degli agenti BPCT di tipo B, per valutare eventuali differenze o somiglianze o coincidenze fra i comportamenti degli agenti cognitivi, a prescindere dal metodo utilizzato per la loro costruzione.

Si vuole, cioè, vedere se è possibile che agenti cognitivi dotati di una struttura di apprendimento completamente diversa siano in grado di sviluppare strategie simili per reagire agli stimoli dell'ambiente.

La struttura generale dei *classifier system* introdotti nel modello Sum è esattamente quella descritta nel paragrafo precedente, secondo lo schema ERA, sono stati creati un *CWRuleMaster* ed un *CWRuleMaker* specifici per gli agenti CW, una *CWInterface*, un *CWDataWareHouse*, che contiene tutte le regole, un file che contiene i parametri del *classifier system* e gli oggetti specifici dei *classifier*, *CWreasury*, *CWMessage*, *CWShuffler*, *CWRule*, *CwEffector* e *CWPriceRuleMaster*.

Nella *CWInterface* sono state codificate le regole per tradurre gli eventi dell'ambiente nella metrica interna dell'agente e, quindi, la Interface contiene una vera e propria chiave di lettura delle regole di comportamento dell'agente CW.

Le regole degli agenti CW, come è stato illustrato nel paragrafo precedente, sono costituite da due parti, una parte rappresenta la <condizione> e codifica i dati ambientali ed una parte che viene

definita < azione>, in cui è codificata la risposta della regola ai dati ambientali, secondo un rapporto di antecedente e conseguente.

La *condition part* delle regole dell'agente CW contiene i dati relativi alle quattro variazioni del prezzo dal giorno (t-5) al giorno (t-1), Il prezzo previsto dall'agente previsore al giorno *nAheadForecasting*, il livello di azioni in portafoglio ed il livello di liquidità, mentre la *action part* contiene le possibili azioni dell'agente sul mercato, che consistono nel vendere o acquistare azioni secondo una quantità che va da uno a tre, l'agente può anche non agire, cioè non comprare e non vendere. L'*effector* suggerisce all'agente quale azione eseguire.

La quantità di azioni e la liquidità sono codificate in base a degli intervalli, l'agente CW, cioè, non sa l'esatto ammontare del proprio patrimonio in denaro ed in quantità di titoli, ma sa se ha delle azioni e se ne ha più di una certa quantità (nell'esperimento base la quantità è 100), allo stesso modo sa se è scoperto di azioni entro lo stesso intervallo, lo stesso vale per la liquidità.

I valori di quantità di azioni e liquidità non sono codificati al loro valore esatto, perchè si avrebbero dei problemi dovuti alla lunghezza delle regole a causa della codifica binaria.

In una prima versione si è tentato di far evolvere regole molto semplici, la cui condizione era costituita semplicemente dalle quattro variazioni di prezzo dei cinque giorni precedenti, gli esperimenti condotti con agenti che rispondevano a queste regole, però, hanno mostrato di non avere molto significato, infatti, tutti i tipi di agenti costruiti con il metodo CW non si sono dimostrati in grado di reagire alle bolle e nemmeno di riuscire ad incrementare il proprio livello di ricchezza, probabilmente per il fatto che, dato che gli agenti possono operare allo scoperto senza limiti, vendere risultava sempre la strategia migliore, per cui evolvevano solo regole che dicevano di vendere ed il *classifier* si stabilizzava su questo tipo di strategia.

La regola che ottiene il diritto di agire, dopo aver pagato una quota del suo patrimonio per poter partecipare all'asta, riceve un *reward* per l'azione compiuta.

Tale premio viene calcolato in modo diverso per i tre tipi di agenti, infatti, mentre gli agenti *CWAgentA* e *CWAgentB*, vengono premiati se incrementano la propria ricchezza calcolata come somma della liquidità e delle azioni al prezzo di chiusura, gli agenti del tipo *CWAgentC* vengono premiati se incrementano la loro ricchezza calcolata come somma della liquidità e delle azioni al prezzo previsto dal *Forecaster*.

Questa differenza è molto importante per lo sviluppo delle regole da parte del *classifier system*, perché se gli agenti *CWAgentA* e *CWAgentB*, dovrebbero imparare ad adeguarsi alle variazioni effettive del mercato, i *CWAgentC* dovrebbero invece imparare a seguire la previsione, si vedrà dai risultati degli esperimenti, riportati nel paragrafo successivo che questa differenza dà dei risultati abbastanza inattesi.

Un'ulteriore differenza fra gli agenti riguarda in particolar modo i *CWAgentB*, i quali corrispondono esattamente ai *CWAgentA*, se non per il fatto che condividono un unico *dataWarehouse*, mentre i *CWAgentA* ed i *CWAgentC* possiedono un *dataWarehouse* per ogni singolo agente.

Gli agenti del tipo B, quindi, vorrebbero rappresentare dei singoli risparmiatori che si rivolgono ad un consulente finanziario per avere consigli sugli investimenti, posto che tali consigli possono

tranquillamente essere diversi per ogni agente, perché dipendono dalle condizioni del mercato nel momento in cui il singolo individuo interroga il *dataWareHouse*.

Questo espediente, che non viene quasi mai utilizzato per gli esperimenti, è servito soprattutto per valutare i tempi di apprendimento dei *classifier* all'interno del programma, infatti, se il meccanismo di evoluzione delle regole viene gestito su un solo magazzino di regole si verifica un apprendimento più rapido e decisamente meno oneroso in termini di tempi computazionali.

Di seguito sono illustrati i parametri del *classifier system* all'interno del modello Sum.

Seme di innesco dei numeri casuali:

randomSeed = 123456;

Parametri Generali:

numberOfRules = 64;  
numberOfEffectors = 7;  
geneLength = 9;  
maxNumberOfMessages = 1;  
effectorsFlag = 1;  
wildCardRate = 0.3;  
confidence = 0.0;  
initialStrength = 10.0;  
shuffleFlag = 1;  
maxMemo = 512;  
evaluationConfidence = 0.0;  
evaluationRate = 0.0;

Parametri per manipolare la forza delle regole:

bidTaxRate = 0.15;  
lifeTaxRate = 0.01;  
bidRatio = 0.25;  
linearBid1 = 0.25;  
linearBid2 = 1.0;  
effectiveLinearBid1 = 0.25;  
effectiveLinearBid2 = 1.0;  
bidSigma = 0.075;  
bidMu = 0.0;

Parametri per gestire il processo evolutivo gestito dall'algoritmo genetico:

evolutionRate = 0.05;  
turnoverRate = 0.5;  
crossoverRate = 1.0;  
mutationRate = 0.001;  
crowdingRate = 0.9;  
crowdingFactor = 0.9;  
verifyNews = 1;

**Fig. 6.6** – Parametri del classifier system per il modello SUM.

I parametri per l'algoritmo evolutivo sono particolarmente delicati da manipolare e modificare, infatti, possono influire in modo determinante sulla ricerca delle soluzioni e, soprattutto sulla bontà delle soluzioni stesse.

Se le regole evolvono secondo parametri di mutazione troppo elevati si può incorrere nel rischio che il classifier system impari troppo in fretta e che la soluzione trovata dal *classifier system* non corrisponda ad una soluzione generale, ma sia solo localizzata e, quindi, non porti ad un comportamento significativo da parte dell'agente che segue le regole del *classifier*.

Perché le regole sviluppate corrispondano, dunque, ad un comportamento significativo dell'agente, occorre che l'apprendimento sia più lento e che il *classifier* abbia la possibilità di esplorare il maggior numero di situazioni possibili, in modo tale da poter imparare a reagire con una certa coerenza agli stimoli del mercato in cui opera.

Qui di seguito è riportato un esempio delle regole sviluppate dal *classifier* per un singolo agente, le regole, si ricorda che sono codificate in base ad un alfabeto binario e sono costituite da nove bit.

Rules by CW\_Agent number 309 after 100 steps are:

```
47F2130 - DataWarehouse cointains following data:
47F42D0 - RuleList contains 64 rules:
47F7FB0 strength: 3.660323 specificity 0.888889
#00#00011
000000110
My partnerList contains 0 entries for:
4996EC0 strength: 3.660323 specificity 0.833333
#1#1#0000
000000101
My partnerList contains 0 entries for:
47FC330 strength: 3.660323 specificity 0.666667
###1##1#1
000000010
My partnerList contains 0 entries for:
49955C0 strength: 3.660323 specificity 0.833333
#1#1#0000
000000100
My partnerList contains 0 entries for:
47F75B0 strength: 3.660323 specificity 0.944444
00010000#
000000000
My partnerList contains 0 entries for:
47D7530 strength: 3.660323 specificity 1.000000
001000011
000000110
My partnerList contains 0 entries for:
47F61B0 strength: 3.660323 specificity 0.777778
#11#0#1#1
000000110
My partnerList contains 0 entries for:
4996C70 strength: 3.660323 specificity 0.888889
#01#00011
000000101
My partnerList contains 0 entries for:
```

```
47F8730 strength: 3.660323 specificity 0.888889
#001#1101
000000101
My partnerList contains 0 entries for:
47F6430 strength: 3.660323 specificity 0.777778
#1#10#1#1
000000001
My partnerList contains 0 entries for:
47FE3B0 strength: 3.660323 specificity 0.888889
#0#000100
000000100
My partnerList contains 0 entries for:
47DED30 strength: 3.660323 specificity 0.888889
#00#00100
000000110
My partnerList contains 0 entries for:
47D6130 strength: 3.660323 specificity 0.833333
0##10#101
000000001
My partnerList contains 0 entries for:
47FA530 strength: 3.660323 specificity 0.833333
1#11#001#
000000101
My partnerList contains 0 entries for:
47F89B0 strength: 3.660323 specificity 0.722222
11####10
000000011
My partnerList contains 0 entries for:
47F5F30 strength: 3.660323 specificity 0.944444
#10110111
000000000
My partnerList contains 0 entries for:
47FD4B0 strength: 3.660323 specificity 1.000000
010000011
000000000
My partnerList contains 0 entries for:
47F66B0 strength: 0.130695 specificity 0.944444
1#010000000100
My partnerList contains 1 entries for:
47F43B0
47F7AB0 strength: 3.660323 specificity 0.944444
1#1110111
000000110
My partnerList contains 0 entries for:
47FAF30 strength: 3.660323 specificity 0.833333
110#000##
000000100
My partnerList contains 0 entries for:
47E49B0 strength: 3.660323 specificity 0.833333
#00#0#111
000000011
My partnerList contains 0 entries for:
47FC5B0 strength: 3.660323 specificity 0.777778
11####000
000000010
My partnerList contains 0 entries for:
47F9130 strength: 3.660323 specificity 0.888889
110011##1
000000100
My partnerList contains 0 entries for:
47FBE30 strength: 3.660323 specificity 0.722222
##0#11##1
000000110
```

My partnerList contains 0 entries for:  
47F6930 strength: 3.660323 specificity 0.888889  
0000#01#1  
000000101  
My partnerList contains 0 entries for:  
47F93B0 strength: 3.660323 specificity 0.722222  
0#1###01#  
000000000  
My partnerList contains 0 entries for:  
47F5530 strength: 3.660323 specificity 0.888889  
11#01#000  
000000011  
My partnerList contains 0 entries for:  
47F7330 strength: 3.660323 specificity 0.833333  
1#00#11#0  
000000100  
My partnerList contains 0 entries for:  
47F8230 strength: 3.660323 specificity 0.833333  
#00111##0  
000000000  
My partnerList contains 0 entries for:  
47FA030 strength: 3.660323 specificity 0.944444  
#00110111  
000000001  
My partnerList contains 0 entries for:  
47FDEB0 strength: 3.660323 specificity 0.722222  
1#1###01#  
000000011  
My partnerList contains 0 entries for:  
47F7830 strength: 3.660323 specificity 0.888889  
0#011#010  
000000100  
My partnerList contains 0 entries for:  
47F5CB0 strength: 3.660323 specificity 0.888889  
1#101#001  
000000010  
My partnerList contains 0 entries for:  
47FCFB0 strength: 3.660323 specificity 0.888889  
010#101#0  
000000001  
My partnerList contains 0 entries for:  
47FC0B0 strength: 3.660323 specificity 0.722222  
#11###01#  
000000010  
My partnerList contains 0 entries for:  
47FA2B0 strength: 3.660323 specificity 0.833333  
1#0#001#0  
000000000  
My partnerList contains 0 entries for:  
47F5A30 strength: 3.660323 specificity 0.777778  
#11#0#00#  
000000011  
My partnerList contains 0 entries for:  
47FEB30 strength: 3.660323 specificity 0.833333  
1#1#01#10  
000000000  
My partnerList contains 0 entries for:  
47FAA30 strength: 3.660323 specificity 0.777778  
0#10###00  
000000110  
My partnerList contains 0 entries for:  
47DD430 strength: 3.660323 specificity 0.777778  
#1#10##11

000000101  
My partnerList contains 0 entries for:  
47FB1B0 strength: 3.660323 specificity 0.777778  
#00#0#1#1  
000000000  
My partnerList contains 0 entries for:  
47FB930 strength: 3.660323 specificity 0.833333  
#1#10#101  
000000001  
My partnerList contains 0 entries for:  
47F4DB0 strength: 3.660323 specificity 0.722222  
11####01  
000000010  
My partnerList contains 0 entries for:  
47F7D30 strength: 3.660323 specificity 0.833333  
#01011##1  
000000101  
My partnerList contains 0 entries for:  
47FD230 strength: 3.660323 specificity 0.777778  
1#110###1  
000000101  
My partnerList contains 0 entries for:  
47FCAB0 strength: 3.660323 specificity 0.722222  
#1#10##1#  
000000101  
My partnerList contains 0 entries for:  
47F57B0 strength: 3.660323 specificity 1.000000  
010110111  
000000011  
My partnerList contains 0 entries for:  
47F84B0 strength: 3.660323 specificity 0.888889  
001011##1  
000000010  
My partnerList contains 0 entries for:  
47F6BB0 strength: 3.660323 specificity 0.944444  
00100001#  
000000000  
My partnerList contains 0 entries for:  
49B4B08 strength: 3.660323 specificity 0.888889  
01#1#0000  
000000001  
My partnerList contains 0 entries for:  
47F70B0 strength: 3.660323 specificity 0.833333  
1#1#01#01  
000000001  
My partnerList contains 0 entries for:  
47FACB0 strength: 3.660323 specificity 0.888889  
0101000##  
000000011  
My partnerList contains 0 entries for:  
49969E0 strength: 3.660323 specificity 0.944444  
010111#00  
000000011  
My partnerList contains 0 entries for:  
47F8EB0 strength: 3.660323 specificity 0.888889  
0#01100#1  
000000100  
My partnerList contains 0 entries for:  
47FC800 strength: 3.660323 specificity 0.666667  
0####01#  
000000110  
My partnerList contains 0 entries for:  
47FA7B0 strength: 3.660323 specificity 1.000000



```
010000100
000000010
My partnerList contains 0 entries for:
  47F9630 strength: 3.660323 specificity 0.833333
11#011##1
000000011
My partnerList contains 0 entries for:
  47FB430 strength: 3.660323 specificity 0.777778
1#1#001##
000000001
My partnerList contains 0 entries for:
  47F98B0 strength: 3.660323 specificity 0.888889
#0010#000
000000110
My partnerList contains 0 entries for:
  47F52B0 strength: 3.660323 specificity 0.833333
#00111##1
000000000
My partnerList contains 0 entries for:
  47FB6B0 strength: 3.660323 specificity 0.833333
#00#0#110
000000001
My partnerList contains 0 entries for:
  47F9DB0 strength: 3.660323 specificity 0.722222
10#00####
000000010
My partnerList contains 0 entries for:
  47FE630 strength: 3.660323 specificity 0.888889
1#101#000
000000010
My partnerList contains 0 entries for:
  47FD730 strength: 3.660323 specificity 0.888889
0000#1#10
000000100
My partnerList contains 0 entries for:

  47F42F0 - EffectorList contains 7 effectors:
  47F4AE0 suggestion 5
000000101
My partnerList contains 0 entries for:
  47F43D8 suggestion 0
000000000
My partnerList contains 0 entries for:
  47F4810 suggestion 3
000000011
My partnerList contains 0 entries for:
  47F4540 suggestion 1
000000001
My partnerList contains 0 entries for:
  47F46A8 suggestion 2
000000010
My partnerList contains 0 entries for:
  47F4978 suggestion 4
000000100
My partnerList contains 1 entries for:
  47F66B0
  47F4C48 suggestion 6
000000110
My partnerList contains 0 entries for:

  47F4310 - MessageList contains 0 messages:

  47F4330 - MatchList contains 1 rules:
```

```

47F4978 suggestion 4
000000100
My partnerList contains 1 entries for:
47F66B0

47F4350 - WinnerList contains 1 rules:
47F66B0 strength: 0.130695 specificity 0.944444
1#010000000100
My partnerList contains 1 entries for:
47F43B0

47F4370 - WorkList contains 0 rules:

47F4390 - MemoList contains 1 messages:
11010
47F43B0 - Treasury balance:
Current amount of treasury is 422.126634

current active effector is:
47F4978 suggestion 4
000000100
My partnerList contains 1 entries for:
47F66B0

Statistics:
Rule's selection has been performed 100 times
CoverDetector has been applied 4 times
CoverEffector has been applied 0 times
Rule's genetic evolution has been performed 6 times

```

### 6.1.3 – Esperimenti, risultati e conclusioni.

L'introduzione degli agenti CWAgent, dei tre diversi tipi, sulla struttura del modello Sum permette di valutare l'ulteriore presenza di agenti cognitivi all'interno di un mercato strutturato.

La creazione degli agenti CW ha seguito diversi passi, si è partiti, infatti, con agenti estremamente semplici, soprattutto per testare il pacchetto CW all'interno di un modello che si può ormai definire come molto completo ed anche complicato e si è via, via cercato di dotarli di un set di informazioni che fosse il più possibile simile a quello degli agenti BPCTB, per poter affrontare un confronto significativo.

Il primo tipo di agenti CW, come si è accennato nel paragrafo precedente era in grado di ricevere dall'ambiente soltanto i dati relativi alle variazioni di prezzo avvenute negli ultimi cinque giorni, in base a questi dati il suo obiettivo era quello di aumentare la propria ricchezza e di riuscire ad imparare come reagire alle bolle speculative che, si è visto, tendono ad emergere nel mercato.

Il risultato ottenuto con questo tipo di agenti è stato decisamente deludente dal punto di vista della reazione alle variazioni del prezzo, mentre ha mostrato un primo risultato apprezzabile per quanto riguarda sia l'aspetto della stabilità del mercato, sia per quanto riguarda la sua prevedibilità.

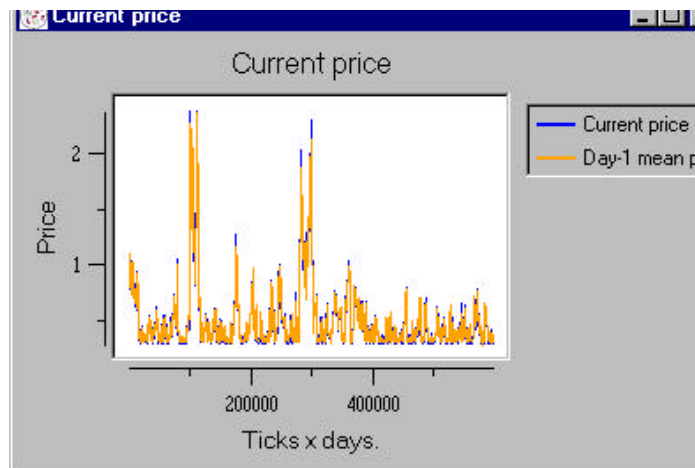


Fig. 6.7 – Grafico del prezzo in un mercato in cui operano 300 agenti, di cui 285 random, 5 cwa, 5cwb e 5cw.

Come si può notare dal grafico, l'andamento del prezzo evidenzia solo due piccole bolle, all'apice delle quali il livello del prezzo raggiunge appena due volte il prezzo iniziale, il mercato, dunque è abbastanza stabile e la percentuale di segni corretti nella previsione raggiunge il 54%.

Il livello della ricchezza degli agenti segue un trend in crescita, ma gli agenti non sono in grado di sviluppare regole sufficientemente valide a non indurli a perdere enormemente durante le bolle.

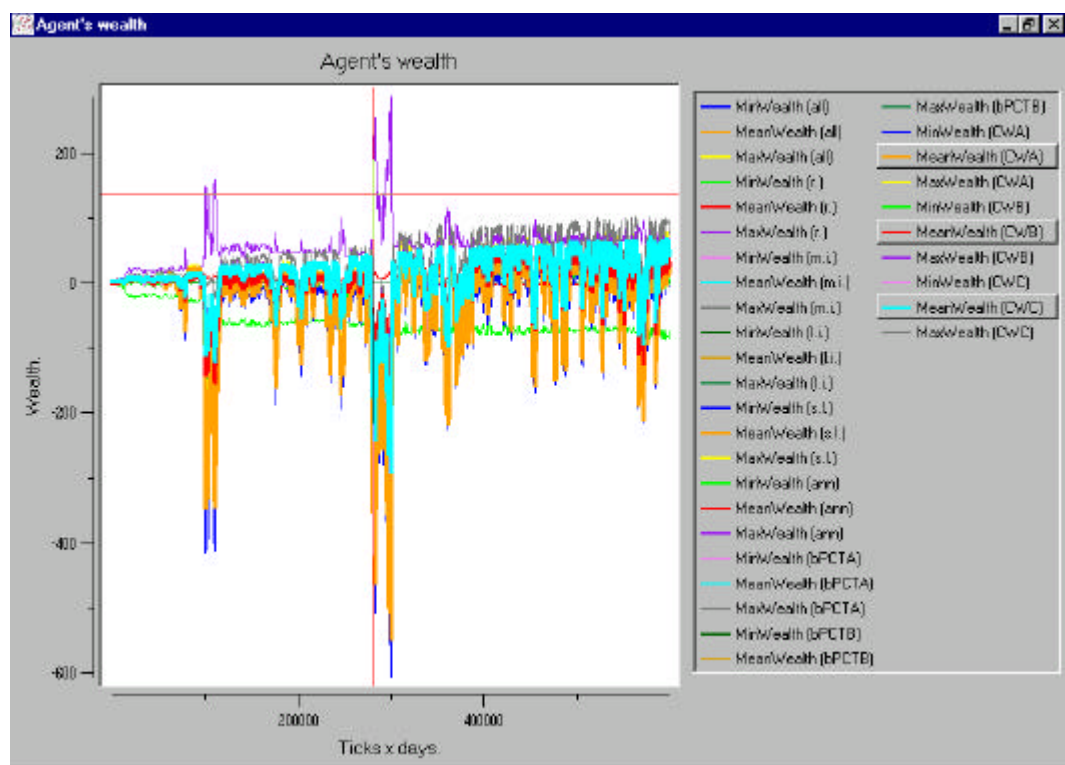
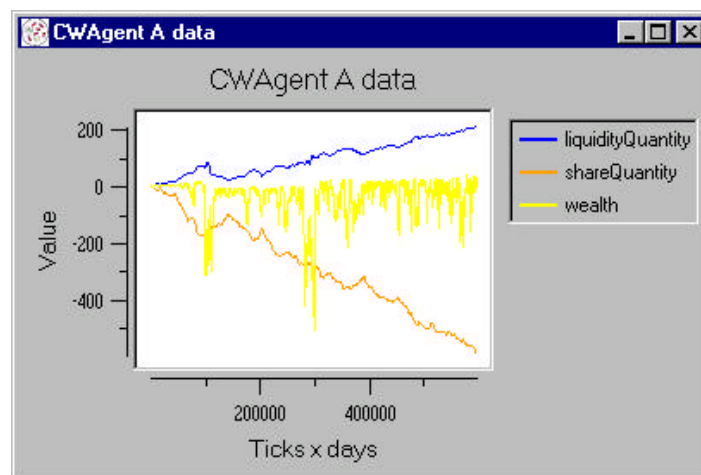


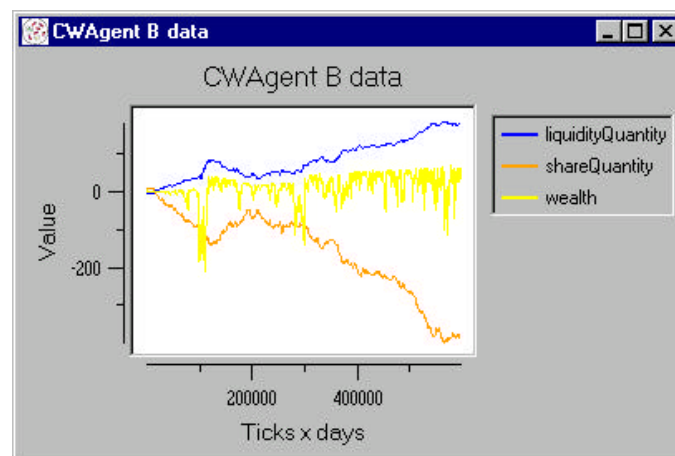
Fig. 6.8 - Grafico della ricchezza in un mercato in cui operano 300 agenti, di cui 285 random, 5 cwa, 5cwb e 5cw.

Come si può notare dal grafico gli agenti del tipo A sono quelli che hanno le maggiori perdite, in valori medi, durante le bolle, mentre gli agenti di tipo B reagiscono leggermente meglio ed i C sembrano essere quelli che presentano la minore varianza nel livello di ricchezza medio.

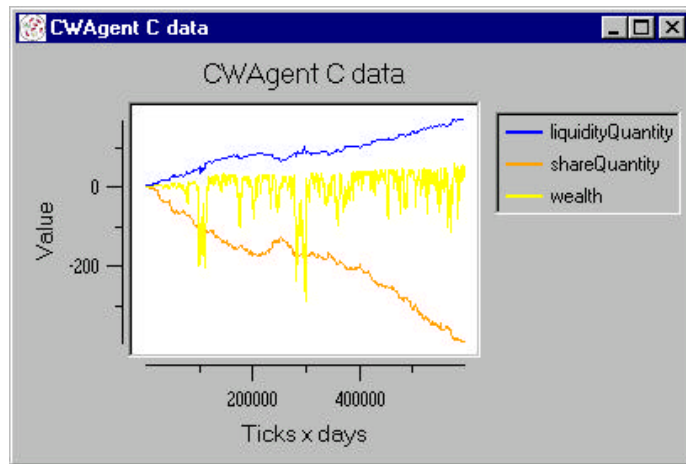
Andando ad analizzare i dati degli agenti si nota che tende ad affermarsi la strategia della vendita, molto probabilmente perché, non avendo alcuna informazione sul proprio patrimonio e potendo operare allo scoperto senza limiti, gli agenti mediamente si arricchiscono, le regole che suggeriscono di vendere diventano subito molto forti e non c'è alcun incentivo a variare la strategia.



**Fig. 6.9** – Dati dei CWAgentA, in cui si evidenzia la quantità di azioni, la liquidità ed il livello medio di ricchezza.



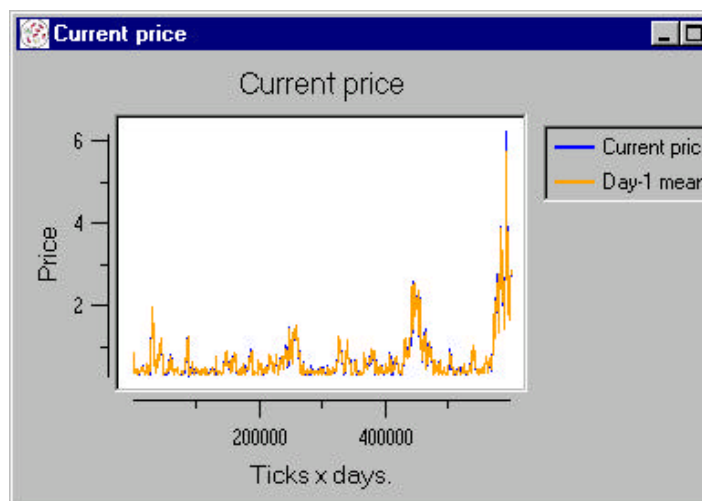
**Fig. 6.10** – Dati dei CWAgentB, in cui si evidenzia la quantità di azioni, la liquidità ed il livello medio di ricchezza, gli agenti di tipo B, imparando più velocemente, mostrano un tentativo di cambiare strategia.



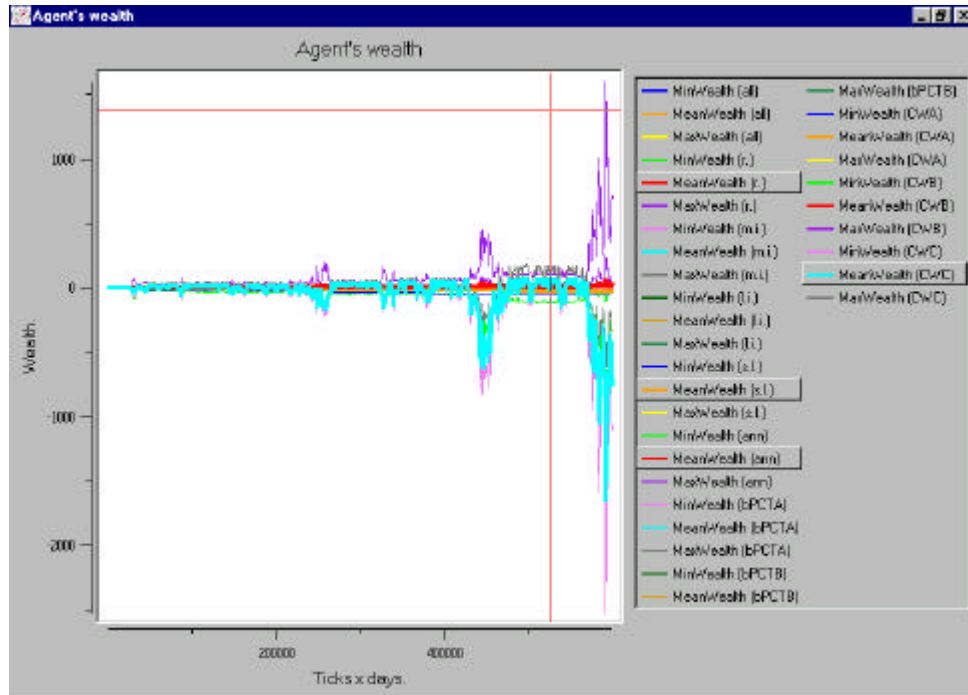
**Fig. 6.10** – Dati dei CWAgentc, in cui si evidenzia la quantità di azioni, la liquidità ed il livello medio di ricchezza, gli agenti di tipo C, che hanno il reward sulla ricchezza calcolata con la previsione, tentano di cambiare strategia, ma tendono a vendere.

Se nel mercato vengono introdotti solo agenti di tipo CWAgentC, lo scenario varia notevolmente rispetto al caso precedente, infatti, il grafico del prezzo evidenzia una sola bolla pari a sei volte il prezzo iniziale, il mercato, quindi è meno stabile ed anche un po' meno prevedibile, infatti la percentuale dei segnali corretti nella previsione scende al 52%.

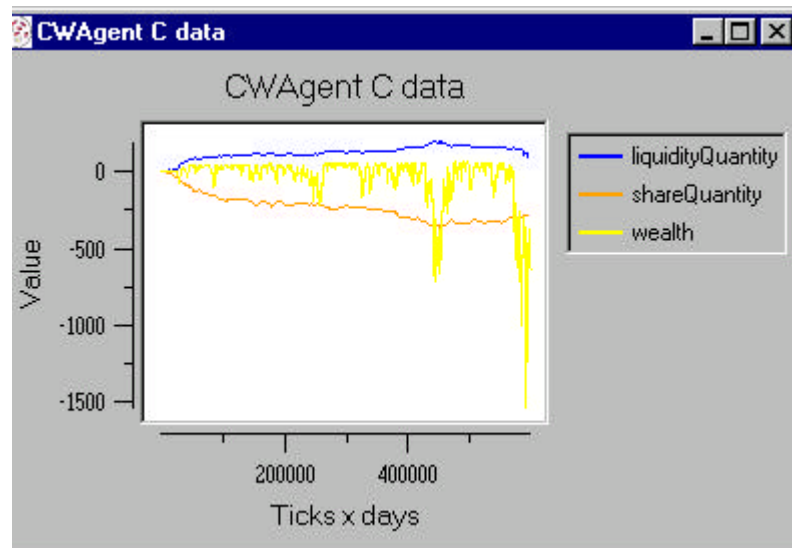
Pur avendo portato il parametro del tasso di mutazione da 0.01 a 0.05, il classifier non è in grado di sviluppare regole tali da superare le bolle, anzi sembra che l'effetto della bolla (in corrispondenza del cinquecento-settantesimo giorno circa) sul livello della ricchezza venga addirittura amplificato.



**Fig. 6.11** – Grafico del prezzo in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp e 5CWAgentC, con mutationRate pari a 0.005.



**Fig. 6.12** – Grafico della ricchezza in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp e 5CWAgentC, con mutationRate pari a 0.005.

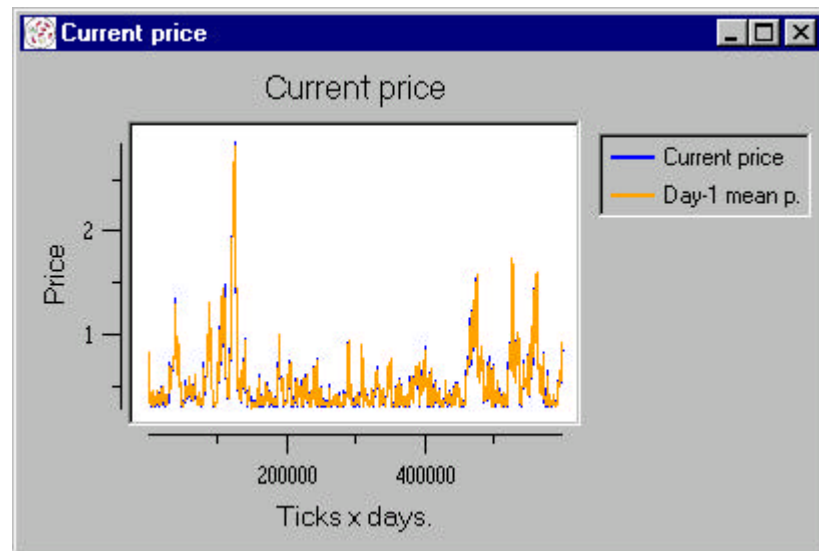


**Fig. 6.13** – Grafico della ricchezza e delle dotazioni degli agenti CWAgentC in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp e 5CWAgentC, con mutationRate pari a 0.005.

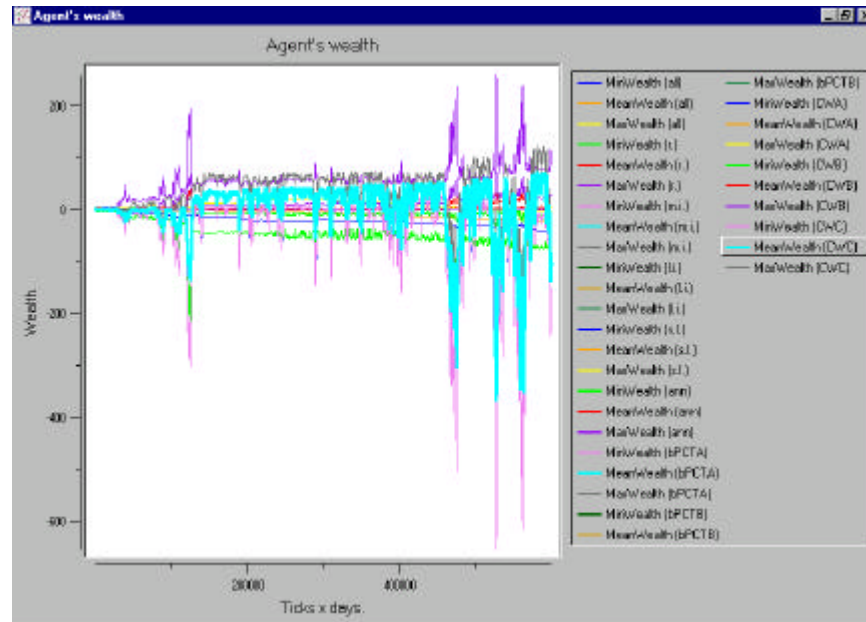
Dato l'effetto molto negativo sul livello della ricchezza media con un mutationRate pari a 0.05, si può provare, analogamente a quanto fatto per l'intensità dell'obiettivo esterno nei BPCTB, a variare ulteriormente il parametro di mutazione, portandolo a 0.03.

In base a questa modifica, peraltro molto semplice anche sotto il punto di vista della gestione del programma, dato che la variazione di un parametro non richiede la ricompilazione del

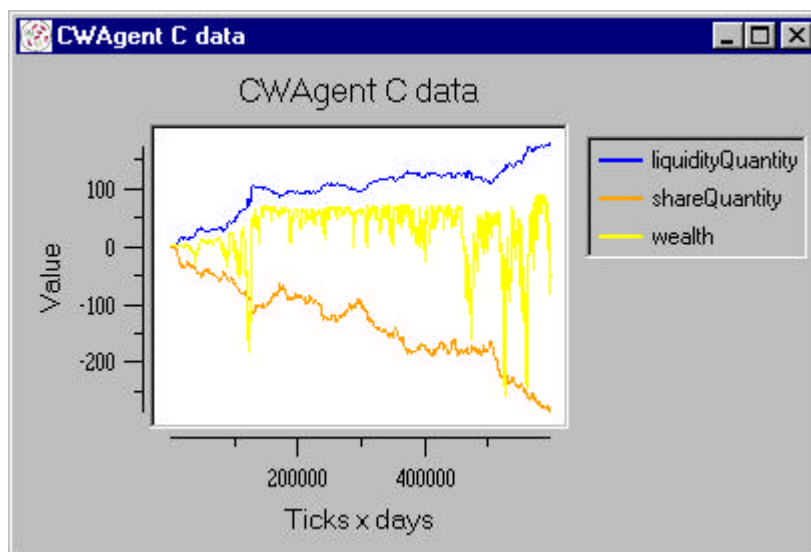
programma stesso, si è riscontrato un aumento della prevedibilità del mercato, che nell'esperimento ha raggiunto il 56% e, contemporaneamente, un aumento della stabilità.



**Fig. 6.14** – Grafico del prezzo in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp e 5CWAgentC, con mutationRate pari a 0.003.



**Fig. 6.15** – Grafico della ricchezza in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp e 5CWAgentC, con mutationRate pari a 0.003.



**Fig. 6.16** – Grafico della ricchezza e delle dotazioni degli agenti in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp e 5CWAgentC, con mutationRate pari a 0.003.

Come evidenziato dalle figure 6.15 e 6.16, gli agenti CWAgentC subiscono ancora le bolle e, in questo caso, è molto evidente che durante il *run* peggiorano sensibilmente, infatti, nonostante le bolle alla fine della simulazione siano quasi inconsistenti, l'effetto sul livello medio della ricchezza è decisamente più significativo rispetto alla prima bolla.

Le strategie degli agenti sono un po' più varie, pur mantenendo la tendenza a vendere ci sono dei momenti, durante la simulazione, nei quali si notano minime inversioni di tendenza.

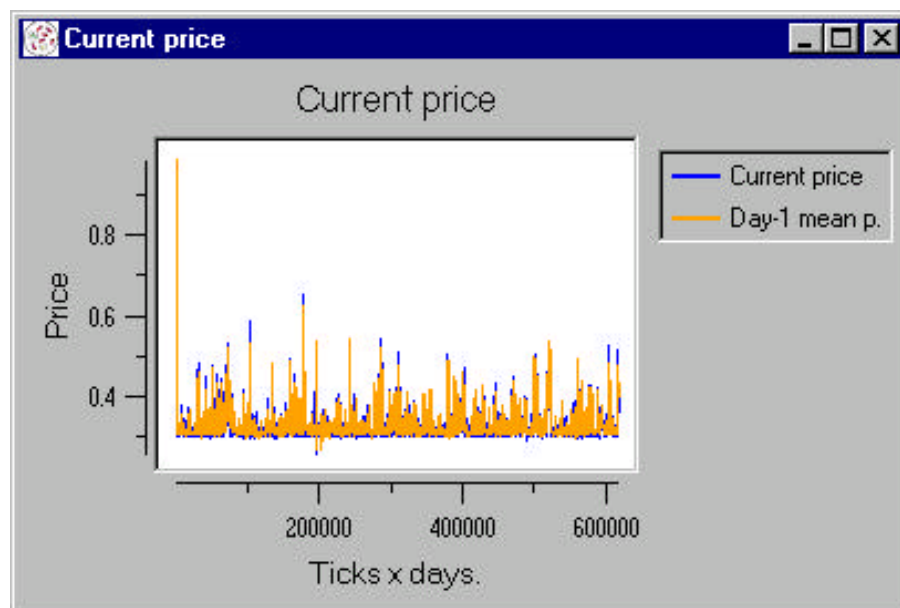
Per ottenere dei risultati migliori occorre, dunque, non agire sui parametri dell'algoritmo genetico del *classifier*, ma sulla struttura vera e propria delle regole, che si sono dimostrate troppo semplici anche se sottoposte ad un *run* molto lungo.

Inserendo nelle regole il valore della quantità di azioni, il valore della liquidità e l'indice di previsione del prezzo futuro, ricevuto dal previsore, si possono apprezzare dei miglioramenti sia per quanto riguarda il mercato in generale, sia per lo sviluppo di strategie "intelligenti" da parte degli agenti.

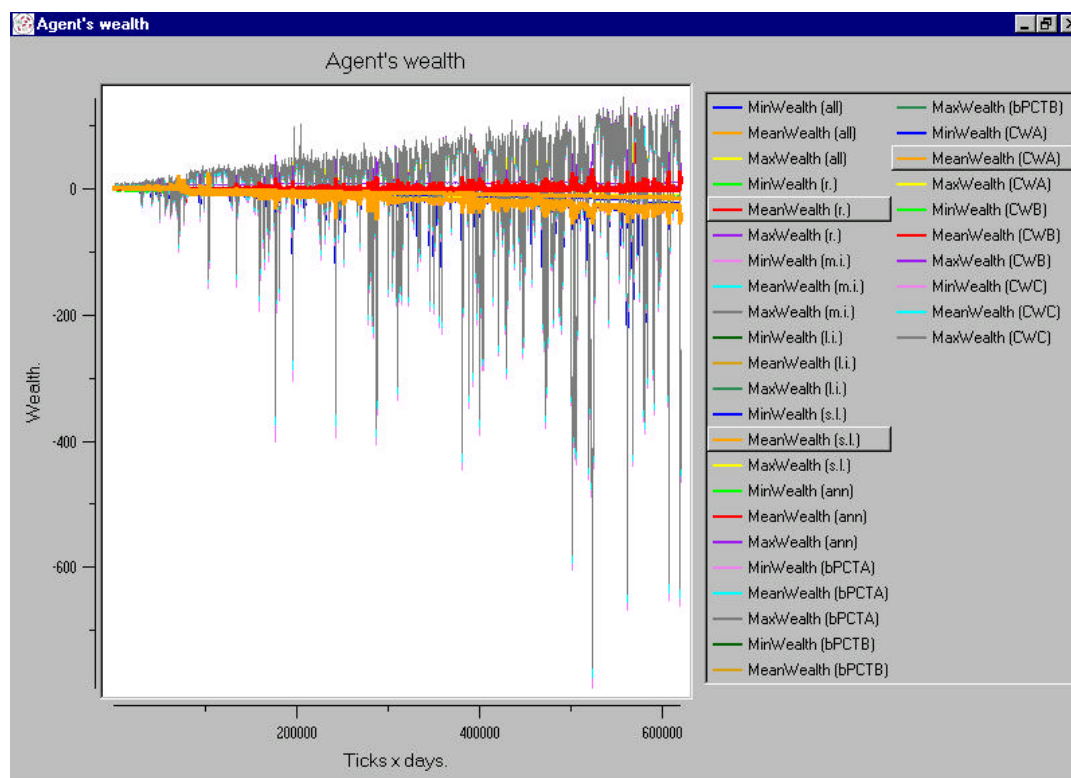
Dal punto di vista del mercato si nota, infatti, una notevole stabilità, il prezzo varia fra 0.2 e 1 e non ci sono bolle evidenti, al tempo stesso si ha quasi il 59% dei segni corretti nella previsione, ci si avvicina ad una situazione di profezie autoavveranti, dovute probabilmente al fatto che gli agenti di tipo CWAgent hanno il valore della previsione in input e ricevono il *reward* sulla ricchezza calcolata in base al prezzo previsto.

Dal punto di vista degli agenti si nota un fatto del tutto inatteso, cioè che mentre gli agenti di tipo C, che ottengono appunto il *reward* sulla previsione, non mutano strategia, ma vendono durante tutto l'esperimento, gli agenti di tipo A mostrano una notevole varietà nelle strategie ed una buona risposta agli stimoli ambientali, riuscendo a reagire minimamente alle bolle.

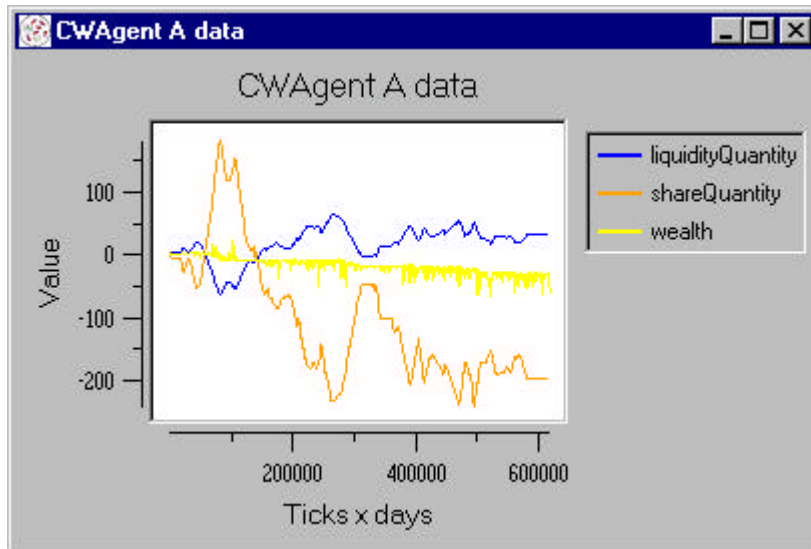




**Fig. 6.17** – Grafico del prezzo in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp 5 CWAgentA, 5CwAgentB e 5CWAgentC con il valore di ricchezza, liquidità e previsione nelle regole.



**Fig. 6.18** – Grafico della ricchezza in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp 5 CWAgentA, 5CwAgentB e 5CWAgentC con il valore di ricchezza, liquidità e previsione nelle regole.



**Fig. 6.19** – Grafico della ricchezza e delle dotazioni degli agenti in un mercato con 270 agenti random, 10 agenti stop loss, 15 aNNForecastApp, 5 CWAgentA, 5CwAgentB e 5CWAagentC con il valore di ricchezza, liquidità e previsione nelle regole.

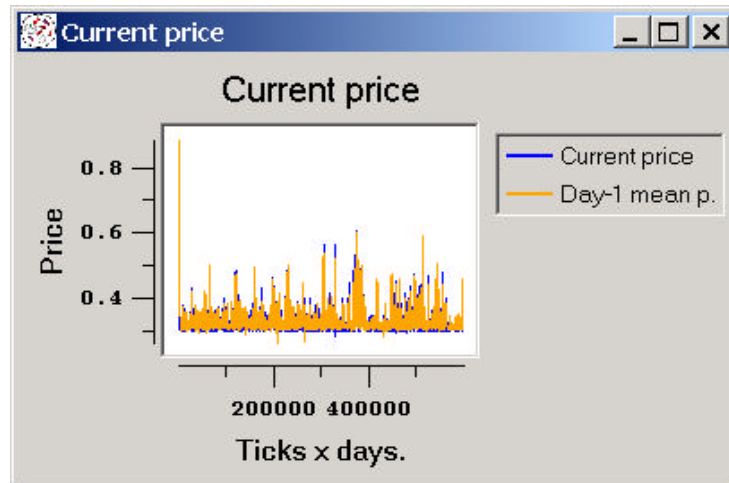
Per evitare che alcune regole si rafforzino troppo velocemente si può operare sui premi che esse ricevono a seguito delle azioni che hanno promosso nell'ambiente.

Introducendo una variazione nell'assegnazione dei premi, non agendo sulla formula della funzione di *reward*, ma sui valori dei premi da assegnare alle regole, cioè riducendo la differenza nei premi fra le regole che hanno agito bene e quelle che hanno agito molto bene da 1 a 0.1, si possono osservare risultati apprezzabili sotto tutti i punti di vista.

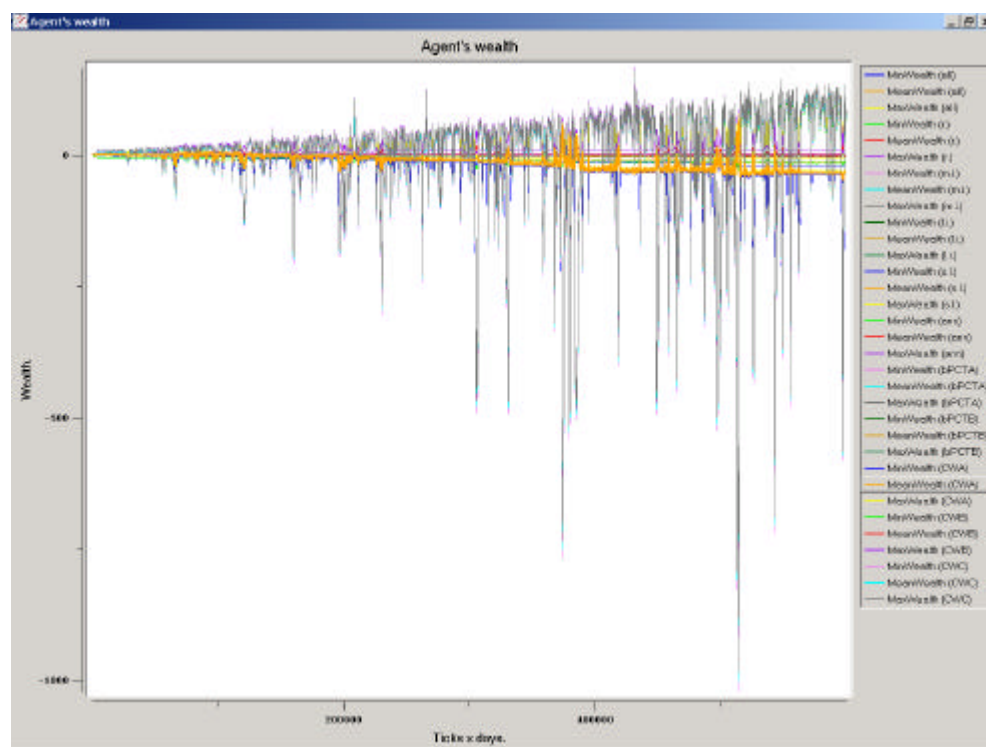
In un mercato in cui operano 260 agenti *random*, 10 *stop loss*, 15 agenti che applicano al previsione della rete neurale e 5 agenti CW del tipo A, del tipo B e del tipo C, si possono riscontrare risultati molto positivi.

Sotto l'aspetto del mercato si può, infatti, notare una notevole stabilità del prezzo, che quasi non presenta bolle, ma resta contenuto fra valori pari a 1 e 0.2.

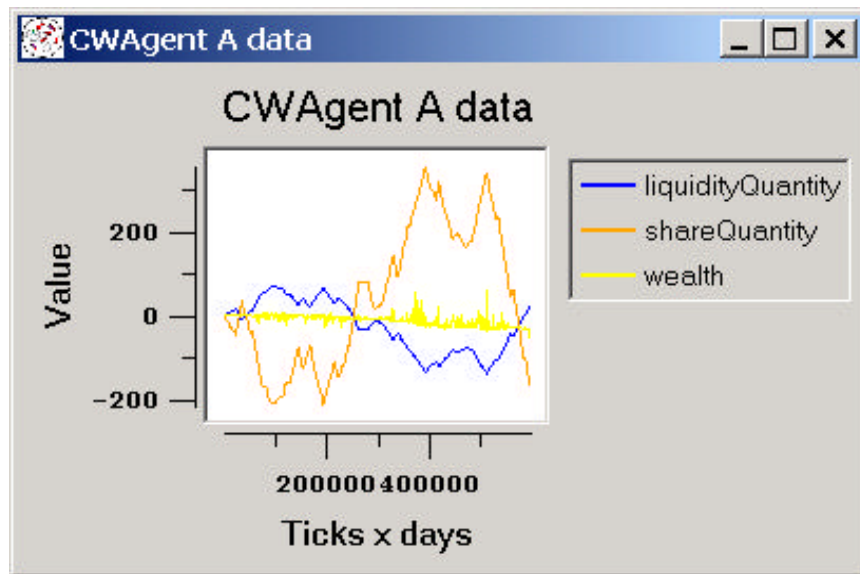
Il livello delle previsioni di segno corretto è del 56% e gli agenti di tipo A presentano non solo una notevole variazione di strategia durante lo svolgimento della simulazione, ma riescono anche a reagire alle piccolissime bolle che si formano ed a mantenere il livello della loro ricchezza media prossimo allo zero.



**Fig. 6.20** – Grafico del prezzo in un mercato in cui operano 260 agenti *random*, 10 *stop loss*, 15 agenti che applicano al previsione della rete neurale e 5 agenti CW del tipo A, del tipo B e del tipo C.



**Fig. 6.21** – Grafico della ricchezza in un mercato in cui operano 260 agenti *random*, 10 *stop loss*, 15 agenti che applicano al previsione della rete neurale e 5 agenti CW del tipo A, del tipo B e del tipo C.



**Fig. 6.22** – Grafico della ricchezza e delle dotazioni degli agenti in un mercato con 260 agenti random, 10 agenti stop loss, 15 aNNForecastApp, 5 CWAgentA, 5CwAgentB e 5CWAgentC.

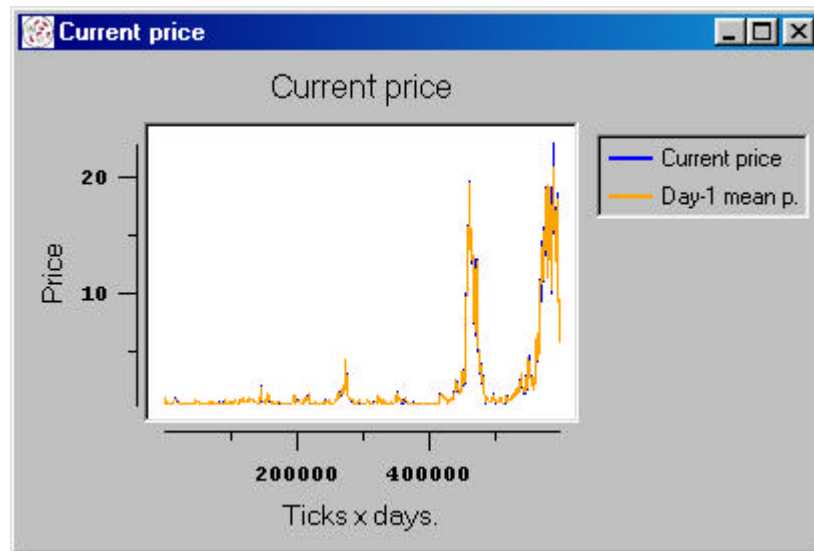
L'ultimo esperimento effettuato introduce due novità nel modello, in primo luogo agli agenti CW, che avevano già in input la previsione fatta dal forecastingAgent, viene indicata l'ampiezza della variazione prevista: gli agenti CW, cioè, sono in grado di individuare se la previsione indica un grande o un piccolo incremento/decremento del prezzo; in secondo luogo, la funzione di reward è stata modificata in modo tale da indurre il classifier ad equilibrare il portafoglio tra la quantità di azioni e la liquidità, in modo tale da avere una maggiore coerenza con i dati in input.

In un mercato con 260 agenti *random*, 10 *stop loss*, 15 *annForecastApp*, 5 agenti CWA, 5 agenti CWB e 5 CWC, si verifica una situazione leggermente diversa rispetto all'esperimento precedente, infatti, la prevedibilità del mercato scende al 48% e il grafico del prezzo mostra due bolle abbastanza considerevoli, all'apice delle quali il prezzo è di circa venti volte il valore iniziale.

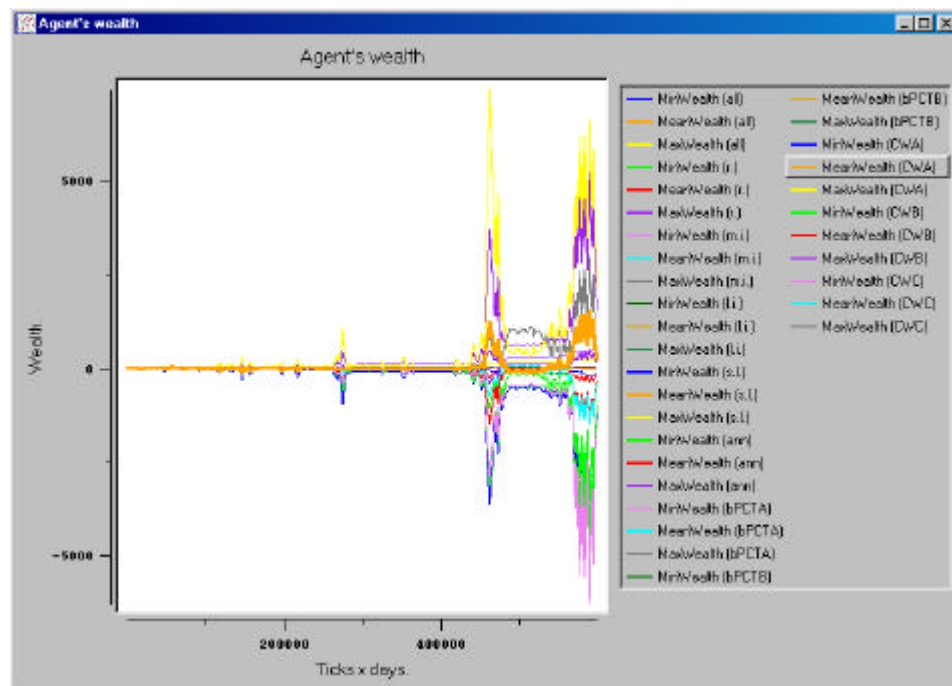
L'aspetto positivo si osserva, però, dal punto di vista degli agenti, infatti gli agenti CW di tipo A, quelli, cioè che ottengono il *reward* in base alla ricchezza effettiva, non solo mostrano una certa varietà di strategie, ma si arricchiscono notevolmente durante le bolle.

Anche gli agenti CW di tipo B, che sono uguali agli A ma condividono un unico *dataWarehouse*, mostrano un comportamento abbastanza positivo, con una notevole varietà di strategie ed una buona reazione alle bolle, anche se la loro ricchezza media durante questi eventi rimane lievemente negativa.

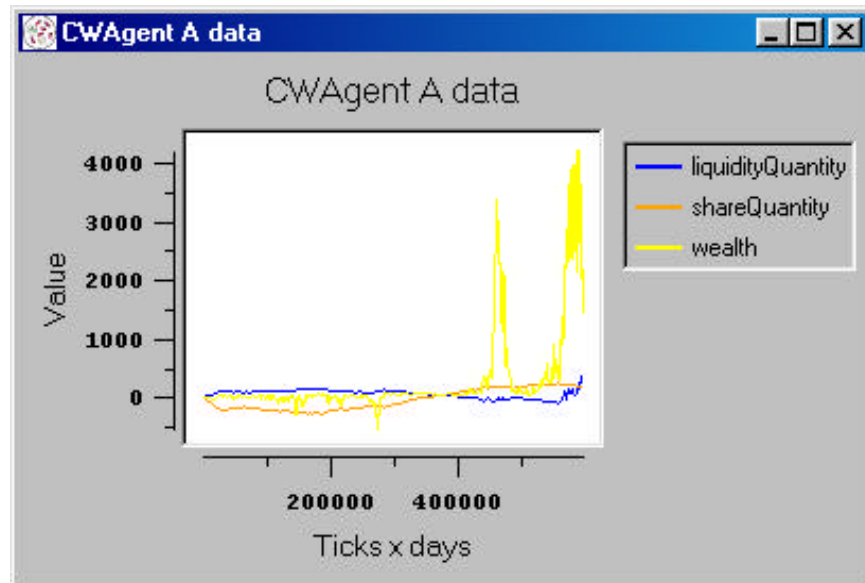
Gli agenti CW del tipo C, invece, continuano ad avere difficoltà a reagire alle bolle, durante le quali tendono a vendere e ad accumulare liquidità.



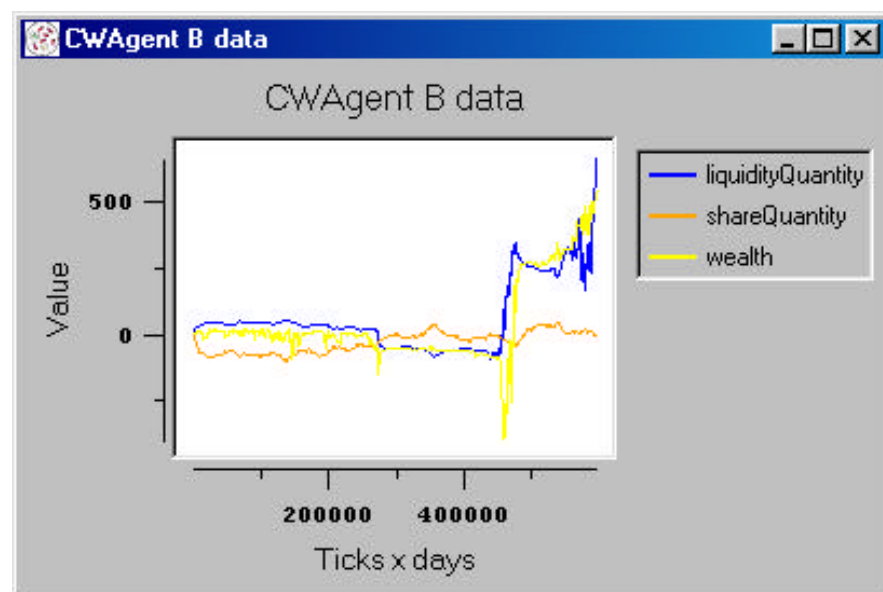
**Fig.6.23** – Grafico del prezzo in un mercato in cui operano 260 agenti *random*, 10 stop loss, 15 aNNForecastApp, 5 CWAgentA, 5 CWAgentB, 5 CWAgentC.



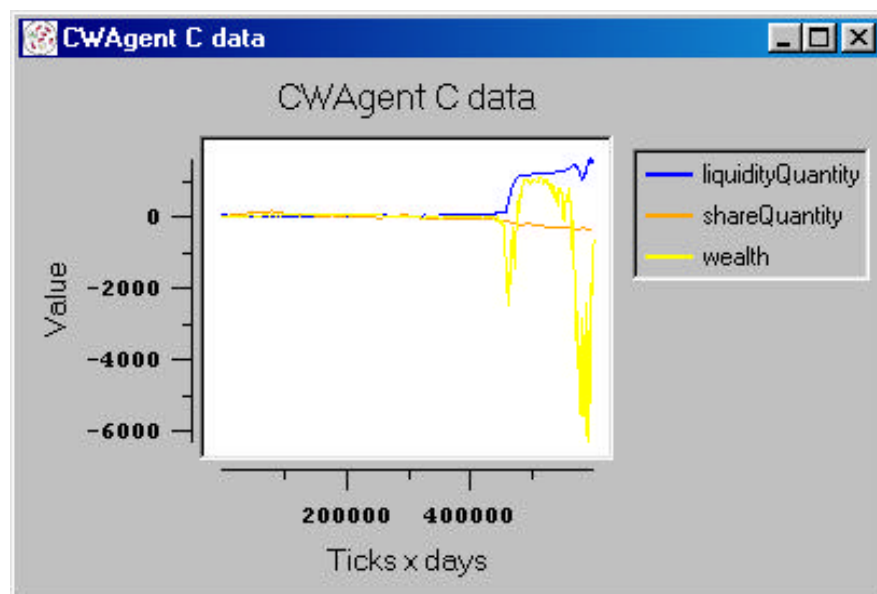
**Fig.6.24** – Grafico della ricchezza in un mercato in cui operano 260 agenti *random*, 10 stop loss, 15 aNNForecastApp, 5 CWAgentA, 5 CWAgentB, 5 CWAgentC.



**Fig.6.25** – Grafico della dotazioni di azioni e liquidità degli agenti CWA, in un mercato in cui operano 260 agenti *random*, 10 stop loss, 15 aNNForecastApp, 5 CWAgentA, 5 CWAgentB, 5 CWAgentC.



**Fig.6.26** – Grafico della dotazioni di azioni e liquidità degli agenti CWB, in un mercato in cui operano 260 agenti *random*, 10 stop loss, 15 aNNForecastApp, 5 CWAgentA, 5 CWAgentB, 5 CWAgentC.



**Fig.6.27** – Grafico della dotazioni di azioni e liquidità degli agenti CWC, in un mercato in cui operano 260 agenti *random*, 10 stop loss, 15 aNNForecastApp, 5 CWAgentA, 5 CWAgentB, 5 CWAgentC.

Gli agenti che operano sviluppando le proprie regole in modo autonomo, grazie all'utilizzo dei *classifier system*, dunque mostrano, di comportarsi in modo molto simile agli agenti cognitivo BPCTB, costruiti con la tecnica dei *CrossTarget*, pur avendo, in alcuni esperimenti, un effetto più stabilizzante all'interno del mercato.

Quando, però si cerca di "indirizzare" il *classifier system* verso un comportamento più simile dal punto di vista strutturale, in un certo senso, ai cross target, agendo sugli input e sugli input e sulla funzione di *reward*, la prevedibilità del mercato scende fino al 48%, e il grafico del prezzo ed il mercato diventano molto più simili alla situazione in cui operano 285 agenti casuali e 15 imitatori del mercato, che non alla situazione di profezie autoavveranti riscontrata in caso do BPCTAgentB con obiettivo esterno di tipo 2 molto forte, apparentemente, quindi, gli agenti cognitivi di tipo CW hanno un effetto sul mercato molto simile a quello dei *chartisti*, il che potrebbe essere giustificato dal fatto che, imparando da esperienze passate, hanno la possibilità di riconoscere delle situazioni e imparano se ripeterle o se comportarsi diversamente, un po' come gli imitatori locali scelgono di imitare o meno il comportamento degli altri agenti, anche se nel loro caso il tutto è dovuto ad una distribuzione di probabilità casuale.

Per quanto riguarda, invece, l'analisi delle cause che generano i fenomeni speculativi delle bolle, anche andando ad analizzare nel dettaglio le regole sviluppate da questo tipo di agenti, non si riesce ad ottenere una risposta esauriente, anche se sono facilmente percorribili i passi che portano gli agenti verso determinati comportamenti.

Probabilmente l'unica risposta plausibile deriva dall'osservazione del *book*, che, durante le bolle presenta lunghe code di ordini senza *match*.

**BIBLIOGRAFIA CAPITOLO SESTO.**

- FERRARIS G. (1999), *Algoritmi Genetici per l'Economia*, Tesi di Laurea presso la Facoltà di Economia dell'Università degli Studi di Torino.



## Bibliografia

- ARTHUR W.B. (1994), *Inductive Reasoning and Bounded Rationality*, in *American Economic Review*, vol.84, n.2, pp. 406-411, Santa Fe Institute, Santa Fe (NM).
- AXTELL R. (2000), *Why Agents? On the Varied Motivations for Agents Computing in the Social Sciences*, In Conte R., Hegselmann R., Terna P. (eds.). *Simulating Social Phenomena, Lecture Notes in Economics and Mathematical Systems* 456, pp.21-40, Berlin, Springer. Reperibile all'url: <http://www.book.edu/es/dynamics>.
- BAK P. , CHEN K. (1991), *Self-organized criticality*, in *Scientific American*, NY, vol.264, n.1.
- BATESON G. (1976), *Verso un'Ecologia della Mente*, Milano, Adelphi.
- BELTRATTI A., MARGARITA S., TERNA P. (1996), *Neural Networks for Economics and Financial Modelling*. London, International Thompson Computer Press.
- BONABEAU E., DORIGO M., THERAULAZ G., (1999), *Swarm Intelligence: from Natural to Artificial Systems*, Oxford University Press.
- BOWER G.H., HILGARD E.R., (1981), *Theory of Learning*, Englewood Cliffs NJ, Prentice Hall.
- CASETTI E., (1980), *Equilibrium Population Partitions between Urban and Agricultural Occupations*, *Geographical Analysis*, 12:1, pp. 47-54.
- CHOMSKY N.(1979), *Saggi Linguistici*, vol.2. *La grammatica generativa trasformativa*, Bollati Boringhieri, Torino.
- CHURCHLAND P.M., CHURCHLAND SMITH P. (1990), *Può una macchina pensare?*, California University San Diego.
- DAY R. (1994), *Complex Economic Dynamic, Volume I: an Introduction to Dynamical Systems and Market Mechanism*. Cambridge, Ma, MIT Press.
- EDGAR, PETERS (1991), *Chaos and Order in the Capital Market*, Ny, Wiley and Son.
- EDMONDS B. (1999), *Modelling Bounded Rationality in Agent-Based Simulations using the Evolution of Mental Models*. in BRENNER T., *Computational Techniques for Modelling Learning in Economics*, Kluwer.
- EPSTEIN J.M., AXTELL R. (1996), *Growing Artificial Societies – Social Science from the Bottom Up*, Ma, Mit press.
- FERRARIS G. (1999), *Algoritmi Genetici per l'Economia*, Tesi di Laurea presso la Facoltà di Economia dell'Università degli Studi di Torino.
- FORRESTER J.W. (1961), *Industrial Dynamics*. Cambridge, MA, Mit Press.
- FRIEDMAN M. (1953), *The Effect of Full Employment Policy on the Economic Stability: A Formal Analysis*, in *Essays in Positive Economics*, Chicago, University of Chicago Press.
- GABER P. (1953), *The Dollar Bubble*, in *Economics of the Dollar Cycle*, a cura di Gerlach S.e Petri P., Cambridge, Ma, MIT Press.
- GLEICK J. (1988), *Chaos: Making a New Science*, NY, Vicking Press.
- HAUGELAND J. (1988), *Intelligenza Artificiale*, Bollati Boringhieri, Torino.
- HAUGELAND J. (1989), *Progettare la Mente. Filosofia, psicologia, intelligenza artificiale*, Il Mulino, Bologna.
- HOBBS T. (1982), *The Leviathan*, (tr. Italiana) *Il Leviatano*, Editori Riuniti, Roma.
- HOLLAND J.H., HOLYOAK K.J., KEITH J., NISBET R.E., THAGARD P.R. (1986), *Induction*, Cambridge (Mass.), Mit Press.
- HOLLAND J.H. (1975), *Adaptation in Natural Artificial Systems*. Ann Arbor,(MI).
- HORGAN J. (1995), *From Complexity to Perplexity*, *Scientific american*, June, 272:6, pp. 104-09.
- HORGAN J. (1997), *The End of Science: Facing the Limits of Knowledge in the Twilight of the Scientific Age*, Paperback ed. New York: Broadway Book.
- JOHNSON P. (2000), *ASM documentation*, reperibile all'url : <http://artstkmkt.sourceforge.net>
- JOHNSON P., LANCASTER A., STEFANSSON B. (1999), *Swarm User Guide*, Santa Fe Institute, New Mexico.
- KALDOR N. (1940), *A Model of the Trade Cycle*, *Economic Journal*. 50 pp.78-92.
- KRUGMAN P.R.(1984), *The International Role of Dollar: Theory and Prospect in Exchange Rate Theory and Practice*. BILSON, JOHN F.O., RICHARD C. MARSTON, eds. Chicago: University of Chicago Press, pp.261-78.
- LANGTON C. (1989), *Artificial Life*. Redwood City, Addison-Wesley.
- LE BARON B.(1991), *Empirical Evidence for Non linearities and Chaos Time Series : a Summary of Recent Results*, University of Wisconsin, Social Systems Research Institute, 9117.
- LEBARON B. (2000), *Agent-Based computational finance: suggested readings and early research*, in *Journal of Economic Dynamics and Control* 24, pp 679-702

- LEBARON B., ARTHUR W.B., PALMER R., (1999), *Time series properties of an artificial stock market*, in Journal of Economic Dynamics and Control 23, pp 1487-1516.
- LOMBARDI M. (1981), *Keynes, ovvero dell'economia non euclidea*, in Studi Economici, 15, pp.76-114.
- LORENZ H. (1989), *Nonlinear Dynamical Economics and Chaotic Motion*, Berlin, Springer-Verlag.
- LUNA F., STEFANSSON B. (2000), *Economic Simulations in Swarm: Agent -Based Modelling and Object Oriented Programming*, Boston, Mass, Kluwer Academic Publisher.
- MANDELBROT B.B., *The Fractal Geometry of Nature*, ny, W.H. Freeman, 1982.
- MARGARITA S. (1992), *Verso un "robot oeconomicus" algoritmi genetici ed economia*, Sistemi Intelligenti, 3, pp. 421-459, Il Mulino.
- MARSHALL A. (1898), *Distribution and Exchange*, Econ.J., 8 (1), 37-59.
- MARSHALL A. (1920), *Principles of Economics*, London, MacMillan Press.
- MINAR N., BURKHART R., LANGTON C., ASKENAZI M. (1996), *The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations*, reperibile all'url: <http://www.santafe.edu/projects/swarm>.
- MORIN E. (1997), *Le vie della Complessità*, in *La sfida della Complessità*, Milano, Feltrinelli.
- MOSS S., EDMONDS B. (1998), *Modelling Economic Learning as Modelling. Systems and Cybernetics*, 29, 5-37.
- PARISI D. (1999), *Mente. I nuovi modelli della vita artificiale*, Il Mulino, Bologna.
- PARISI D. (2001), *Simulazioni. La realtà rifatta nel Computer*, Bologna, Il Mulino.
- ROSSER J.B.JR (1991), *From Catastrophe to Chaos: A General Theory of Economics Discontinuities*, Boston, Kluwer Academic Publishers.
- ROSSER J.B.JR (1999), *On the Complexities of Complex Economic Dynamics*, in *Journal of Economic Perspectives*, Vol.13, num.4, pp.169-192.
- RUELL D. (1991), *Chaotic Evolution and Strange Attractors*, Cambridge, Cambridge University Press.
- RUMELHART D. (1980), *Schemata: the Building Blocks of Cognition*, IN SPIRO R., BRUCE B., BREWER W., *Theoretical Issues in Reading Comprehension*, Hillside (NJ), Erlbaum.
- RUST J. (1996), *Dealing with the Complexity of Economic Calculations*. Working Paper. Department of Economics, Yale University: New Haven, Connecticut.
- SAMUELSON P.A. (1973), *Fondamenti di analisi Economica*, Milano, Il Saggiatore.
- SARGENT T.J. (1993), *Bounded Rationality in Macroeconomics*. Oxford, Clarendon Press.
- SAVAGE L.J. (1972), [1954], *The Foundation of Statistic*, N.Y., Dover Publications.
- SCHANK R., ABELSON R.P. (1977), *Scripts, Plans and Understanding: an Inquiry into Human Knowledge Structures*, Hillside (NJ), Erlbaum.
- SEARLE J. (1980), *Menti, cervelli e programmi*, Berkley.
- SIMON H.A. (1985), *Le teorie della Razionalità Limitata*, in Id. *Causalità, Razionalità ed Organizzazione*, Bologna, Il Mulino.
- TERNA P. (1995), *Paper XXXV Riunione scientifica annuale della Società Italiana degli Economisti*.
- TERNA P. (1998), *Creare mondi artificiali: una nota su Sugarscape e due commenti*, Sistemi Intelligenti, 3-98, pp. 489-496, Bologna, Il Mulino.
- TERNA P. (2000), *The "mind or no mind" dilemma in agents behaving in a market*, in G. Ballot e G. Weisbuch (eds.), *Applications of Simulation to Social Sciences*. Paris, Hermes Science Publications.
- TERNA P. (2001), *Cognitive agents behaviors in a simple stock market structure*. In corso di pubblicazione.
- TESTFATSION L. (1997), *Old Web Site for Agent-Based Computational Economics (ACE)*, reperibile all'url: <http://www.econ.iastate.edu>.
- TESTFATSION L. (1998), *The Labor Market: a Simulative Perspective*. (reperibile al seguente url: <http://www.econ.iastate.edu/testfatsi/evlab.ps>).
- THOM R. (1975), *Structural Stability and Morphogenesis*.
- THOMPSON J.M.T. (1986), STEWART H.B., *Nonlinear Dynamics and Chaos*, NY, Wiley and Sons.
- TURING A. (1994), *Macchine Calcolatrici e Intelligenza*, IN V. SOMENZI, R. CORDESCI, *La Filosofia degli Automi*, Bollati Boringhieri, Torino.
- VARIAN H.R. (1979), *Catastrophe Theory and the Business Cycle*, Economic inquiry, January, pp.14-28.
- WALDROP M. M. (1995), *Complessità, Uomini ed Idee al confine tra Ordine e Caos*, Torino, Instar Libri.
- WEIZENBAUM J. (1987), ***Il potere del computer e la ragione umana***, Gruppo Abele, Torino.
- ZAHLER R. E SUSSMAN H., (1977), *Claims and Accomplishments of Applied Catastrophe Theory*, Nature 269, pp.759-63.

- ZEEMAN E.C. (1974), *On the Unstable Behavior of the Stock Exchanges*, *Jornal of Mathematical Economics*, 1:1, pp.39-44.