

PIETRO TERNA E RICCARDO TAORMINA

## MODELLI DI SIMULAZIONE CON AGENTI INTELLIGENTI: IL SORPRENDENTE MONDO DEI CAMALEONTI

Un percorso di sviluppo della ricerca sugli agenti per le scienze sociali, con particolare rilievo per l'economia, è stato quello, all'inizio degli anni '90, di acquisire lo strumento delle reti neurali per dare forma agli agenti, inizialmente anche trattati in modo isolato. A metà degli anni '90, o poco dopo, si è diffusa nelle scienze sociali la modellistica ad agenti, con capacità di azione e interazione, in particolare grazie alla comparsa di *Swarm* ([www.swarm.org](http://www.swarm.org)) e del protocollo che propone. In parallelo, si sono diffusi i dubbi sulla plausibilità dei modelli neurali artificiali per gli agenti. Ora è il momento di considerare soprattutto una sintesi tra reti neurali e modelli ad agenti. Qui si propongono elementi per questa sintesi, con un esempio fondato sulla metafora dei camaleonti e con la proposta dell'uso del protocollo di *Swarm* all'interno di Python ([www.python.org](http://www.python.org)), potente strumento di programmazione relativamente facile da apprendere, con *SLAPP* (Swarm Like Agent Protocol in Python).

### AGENTI, RETI NEURALI E MODELLI: IL CONTESTO DI RIFERIMENTO

Il contesto cui sono qui riferiti l'uso delle reti neurali artificiali e dei modelli ad agenti è quello della simulazione nelle scienze sociali, intesa come nuova via per la costruzione di modelli (Terna *et al.*, 2006).

I modelli, quali rappresentazione semplificata della realtà (tutti conoscono i modelli, avendo giocato con un'automobilina o con una bambola), possono essere costruiti in molti modi: da quello fisico; alla pura rappresentazione letteraria e descrittiva, infinitamente flessibile; alla costruzione matematico-statistica, con il pregio della possibilità del calcolo; alla simulazione ad agenti.

La prima modalità, fisica, è quella della riproduzione in scala di un oggetto da studiare; la seconda, letteraria, è tipicamente rappresentata da un trattato che riporti, ovviamente non in scala uno a uno, gli accadimenti di un determinato periodo, i legami di causa ad effetto individuati dall'autore, gli approfondimenti scelti e così via. Si tratta certamente di un modello, infinitamente flessibile, ma non computabile. La terza strada, dei modelli fondati sulle equazioni, unisce rigore e possibilità di calcolo, ma con la necessità di dover stilizzare significativamente le situazioni rappresentate. Ad esempio, con l'adozione del cosiddetto agente rappresentativo, che incorpora in sé le caratteristiche standard degli agenti che popolano un determinato ambiente economico ed è strutturato con le necessarie dosi di complicatezza e razionalità che consentono di riprodurre un determinato contesto economico. Quanto più è complicato quest'ultimo, tanto più dovranno essere articolate le regole dell'agente rappresentativo, con l'ipotesi di razionalità olimpica (o, raramente, quelle di razionalità limitata) chiamate a sorreggere

la sua adesione a quelle regole. Si tratta di una strada praticata dall'economia detta ortodossa, indipendentemente dalle critiche, e non solo recenti, anche interne alla disciplina (Kirman, 1992).

La ricerca di nuovi paradigmi economici impone la rottura di questa rigida relazione tra la complicatezza dell'ambiente e quella del modello di agente utilizzato: con la quarta strada, quella della simulazione ad agenti, l'attenzione è puntata su routine informatiche cui si chiede di riprodurre il comportamento degli individui che popolano l'ambiente che si intende simulare. Gli agenti così costruiti, ed il mondo simulato in cui operano, possono quindi essere descritti con una formalizzazione computabile (tramite il linguaggio o l'ambiente informatico adottato), simultaneamente con rigore e flessibilità.

Dalla simulazione ad agenti si ritorna alla simulazione intesa come calcolo di equazioni, in particolare di equazioni alle differenze, quando un sistema sociale possa essere descritto con quelle equazioni e una scrittura ad agenti sia soltanto un modo per rappresentarle o calcolarne la soluzione in mancanza di una via analitica.

Quelle che qui interessano sono invece le situazioni in cui non sia possibile trattare le equazioni che descrivono il sistema (che pur sempre *ex post* esistono, ma non necessariamente *ex ante*) e quindi la formalizzazione ad agenti rappresenti l'unica strada realisticamente percorribile. Un esempio: si supponga di voler verificare a priori l'effetto dell'introduzione dell'asta di chiusura sull'andamento della serie di prezzi della borsa italiana (in effetti in Italia è stata introdotta da alcuni anni: per evitare comportamenti erratici o maliziosi da parte di alcuni operatori in fine di giornata, il mercato è strettamente regolamentato in chiusura, come lo è in apertura). Possiamo costruire un modello di simulazione con agenti che operino in sequenza temporale, in tempo discreto, secondo regole proprie, molto o poco complicate, con composizioni diverse della numerosità dei gruppi di agenti, e farli interagire con il sistema telematico della borsa che accoda gli ordini di acquisto e di vendita (il cosiddetto book) secondo le regole operanti in Italia, sia con l'asta di chiusura, sia senza. Le diverse serie di prezzi così generate, a parità di ogni altra condizione del sistema di simulazione, si differenziano per la presenza o l'assenza dell'asta di chiusura; il rilievo che quest'ultima assume sarà maggiore o minore a seconda del tipo di agenti che operano nella simulazione, permettendo di valutare l'utilità e l'effettiva operatività dello strumento introdotto, o che si vuole introdurre, nel sistema borsistico, a seconda delle caratteristiche degli operatori. In questo caso le equazioni *ex ante* del problema potrebbero essere scritte con molta difficoltà e solo sulla base di robuste semplificazioni, che invece la simulazione ad agenti non richiede.

Rispetto alle rappresentazioni ad equazioni, nei modelli ad agenti è preminente l'attenzione all'aspetto descrittivo della realtà e alle microanalisi fondate sugli individui, tenendo conto della loro eterogeneità e delle possibili interazioni (è il caso della borsa).

Grazie alla capacità descrittiva flessibile di un codice informatico scritto ad agenti realizziamo dunque esperimenti mentali assistiti dal computer, in quanto troppo complessi per essere risolti senza l'ausilio del calcolo automatico. Ciò consente di verificare la conseguenza delle caratteristiche e delle capacità attribuite agli agenti nel

contesto in cui operano. Una delle «scoperte» frequenti è infatti proprio il legame tra agenti semplici e risultati complessi.

Abbiamo dunque situazioni complesse quando l'azione e interazione dei soggetti produce risultati non previsti, o addirittura non prevedibili, rispetto ai singoli componenti.

Siamo comunque lontani dalle più classiche argomentazioni dell'individualismo metodologico, secondo cui i concetti sociali collettivi devono essere sempre esplicitamente definibili per mezzo di termini individuali, perché tra la somma e le sue parti scopriamo la possibile presenza della complessità, come il pensiero hayekiano prefigurò anzitempo, come riportato più avanti nel paragrafo su complessità e emergenza. Su questi temi vedere anche Conte e Terna (2000).

Un campo molto prossimo è la ricerca sull'intelligenza degli sciami (Bonabeau *e al.* 1999), in cui si presta attenzione soprattutto a sistemi biologici (insetti collettivi) e a sistemi robotici. L'analisi può però essere estesa anche a sistemi sociali. Le capacità dei sistemi studiati hanno infatti la caratteristica di trascendere l'abilità dei singoli individui costituenti: in molti casi si rilevano comportamenti di gruppo affidabili ed efficaci, che derivano da semplici regole di interazione tra individui o tra individui ed ambiente.

Si tratta dunque di studiare l'azione dei soggetti economici (anche in termini cognitivi) e contemporaneamente di verificare, con la simulazione, quali siano le conseguenze di ciò che si conosce, o si suppone di conoscere, se si tengono in conto le azioni degli, e le interazioni tra, gli agenti; se si considera l'imitazione; se si considera l'apprendimento. Chiamando in causa gli aspetti cognitivi, si può passare dalla costruzione di agenti semplici a quella di agenti più sofisticati.

La prospettiva è di confrontarsi con la complessità, cioè con la non linearità dei comportamenti aggregati rispetto all'azione degli agenti, con l'apparente contrapposizione tra formicaio e formiche (straordinaria complessità aggregata e semplicità delle componenti) o tra «fenomeni economici» ed «azione delle persone»; si ottengono così risultati anche inattesi sotto forma di emergenza di fenomeni, regolarità e strutture; infine si possono proporre contesti che non conoscono stati di equilibrio o ne conoscono molti o sono in transizione verso un equilibrio.

Riassumendo e seguendo l'impostazione di un recente libro sul tema della simulazione sociale (Gilbert e Troitzsch, 2005), la simulazione ad agenti è dunque una particolare tipologia di modelli. La costruzione di un modello rappresenta una strada consolidata per comprendere il mondo: un modello è una semplificazione di una struttura esistente. Passando dal modello al programma informatico che lo rappresenta, una volta che la teoria è formalizzata in un codice di calcolo e noi abbiamo stabilito alcuni parametri di avvio, il codice può essere eseguito e conseguentemente noi possiamo osservare il comportamento della simulazione. Il computer diventa così il nostro laboratorio artificiale, ma collegato ad una realtà che noi consideriamo naturale. Per questo motivo si fa anche riferimento a questa metodologia con l'espressione «esperimento artificiale».

La simulazione ad agenti è anche una promettente strada per rappresentare la realtà: catturare la realtà e farne un modello «nel computer» significa proporre a sé e agli altri studiosi un punto di partenza di grande valore e rigore; certo per fare ciò in modo utile è necessaria una buona dose di realismo della rappresentazione, senza eccedere in fatti e comportamenti stilizzati.

#### AGENTI, RETI NEURALI E MODELLI: METODOLOGIE DI SVILUPPO

Esaminiamo ora come trasformare le idee della simulazione ad agenti per le scienze sociali in un codice concretamente eseguibile. Una proposta è quella di costruire i nostri modelli secondo uno schema che può essere sintetizzato nello schema<sup>1</sup> ERA (Gilbert e Terna, 2000) riportato nella fig.1, gestendo quattro diversi strati nella costruzione del modello e degli agenti.

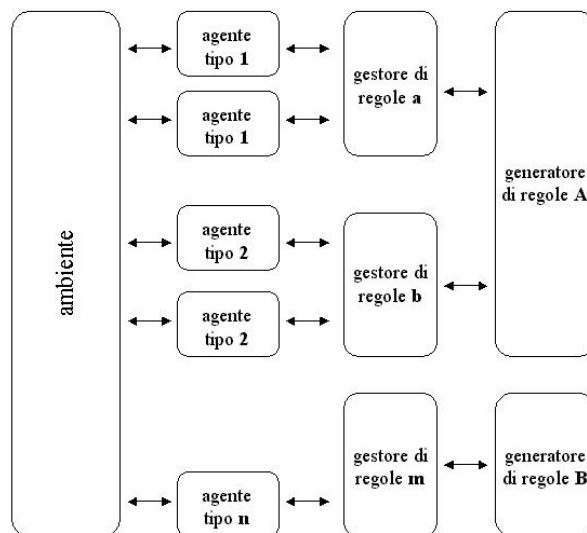


Fig. 1. Lo schema ERA (*Environment, Rules, Agents*).

- Un primo strato rappresenta l'ambiente in cui gli elementi sono chiamati ad interagire. Ad esempio all'interno del protocollo di *Swarm* corrisponde al Modello *Swarm*, vale a dire il contesto all'interno del quale si definiscono gli agenti, si individuano gli eventi nel tempo, si chiariscono le regole di interazione tra gli agenti grazie ai metodi (interpretabili come messaggi che gli agenti sono in grado di gestire, anche reagendo con azioni e produzione di informazioni) definiti all'interno degli oggetti creati dal modello.

<sup>1</sup> Vedere anche <http://web.econ.unito.it/terna/ct-era/ct-era.html>.

- Un secondo strato è quello degli agenti, che possono essere costruiti come esemplari di una o di più classi, a loro volta generate ereditando proprietà, caratteristiche, dati e metodi da classi più generali.
- Il terzo strato gestisce le modalità attraverso cui gli agenti decidono il proprio comportamento. Ad ogni scelta, l'agente interroga un oggetto sovraordinato, definito gestore di regole, comunicandogli i dati necessari ed ottenendo le indicazioni di azione. Sono entrambi strati, questo e il precedente, che possono essere definiti come doppio strato degli agenti.
- Il quarto strato tratta la costruzione delle regole. Esattamente come gli agenti interrogano i gestori di regole, i gestori di regole - per modificare la propria linea di azione - interrogano i generatori di regole. Continuando la considerazione finale al punto precedente, questo può essere definito uno strato di super-agenti.

Lo schema rappresenta sia un tentativo di rendere rigorosamente ordinato il codice informatico di una simulazione, sia una scelta nella direzione della modularità. Se il codice è stato scritto secondo le modalità descritte, è infatti agevole introdurre a volta a volta gestori di regole con caratteristiche diverse, semplicemente sostituendo gli oggetti utilizzati nel modello. Ad esempio, utilizzando come gestore di regole: un sistema a regole fisse (senza generatore di regole, in questo caso); una rete neurale; un algoritmo genetico; un sistema a classificatore (*classifier system*).

A loro volta reti neurali, algoritmi genetici, sistemi a classificatore avranno la necessità di ricorrere a generatori di regole, differenziati a seconda della loro rispettiva tipologia: ciò sarà facilitato dalla modularità del disegno adottato.

Nella prospettiva dell'impiego di semplici reti neurali artificiali *feedforward* per gestire e produrre regole, avremo il passo *forward* (che dato un vettore di input produce un vettore di output) nel terzo strato e una applicazione dell'apprendimento tramite *backpropagation* dell'errore nel quarto strato. A proposito delle reti neurali, si veda l'appendice A per gli aspetti più tecnici

#### AGENTI E RETI NEURALI: UN PROBLEMA DI PLAUSIBILITÀ

Per fare uso di reti neurali artificiali nella costruzione di modelli economici, allo scopo di determinare l'azione degli agenti artificiali in quel contesto, come ad esempio in Terna (2000), occorre però avere affrontato l'aspetto della plausibilità dell'uso dello strumento rete. Le reti neurali artificiali sono funzioni a molti gradi di libertà in grado di approssimare qualsiasi funzione data (Hornik e al., 1990), purché i parametri della rete siano determinati sulla base di una statistica contenente i valori di input e di output della funzione.

Come annota Zorzi (2006), nello studio della cognizione umana i modelli connessionistici rappresentano un notevole passo avanti per quel che riguarda la capacità di simulare il comportamento umano sia in termini qualitativi, sia quantitativi.

Tuttavia i modelli connessionistici sin qui sviluppati si presentano come implausibili, sia per l'architettura neurale semplificata, sia per le tecniche di apprendimento utilizzate, in particolare la classica *backpropagation* applicata a reti *feedforward*. Molti ricercatori hanno scelto di ignorare il problema, dato che la *backpropagation* è un algoritmo potente. Oltre alla implausibilità del meccanismo in sé, questa direzione di applicazione impone l'esigenza continua di un insegnante esterno, ossia di un vettore di output desiderato, per il calcolo dell'errore con cui correggere i parametri della rete (l'errore qui è definito in termini puramente statistici di scarto tra valori attesi e effettivi nel calcolo effettuato dalla rete; non si tratta di errore di scelta, comportamento e così via, che riserviamo all'agente strutturato con una rete o altro sistema di regole). Ma quel vettore in molti casi può non esistere, se quel che vogliamo emerga dalla simulazione è proprio il comportamento degli agenti.

#### IL METODO DEI *CROSS-TARGETS* (CT)

Nel caso di agenti di cui vogliamo scoprire sia un comportamento possibile, sia le relative conseguenze aggregate, sempre all'interno di un modello di simulazione, manca l'insegnante esterno, cioè l'insieme *ex ante* di dati cui adattare la funzione a rete neurale: i dati sono generati solo all'interno del modello, dall'agente che opera.

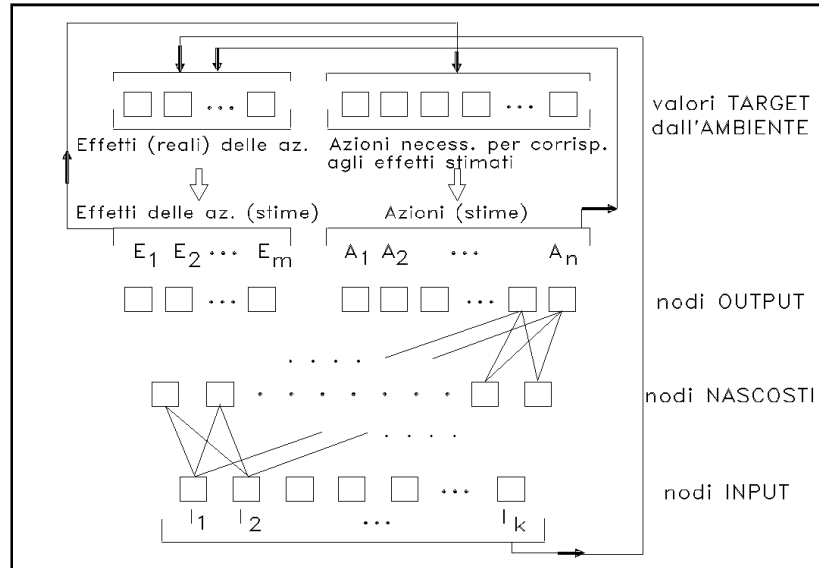


Fig. 2. Lo schema *Cross-Targets* (CT).

Per superare questo problema (resta l'altro aspetto non plausibile segnalato da Zorzi, vale a dire l'uso dell'algoritmo della *backpropagation* per l'apprendimento) propongo la metodologia detta dei *Cross-Targets*, presentata inizialmente in Beltratti e al. (1996). Nella fig. 2 se ne descrive l'impostazione esplicitamente connessionista.

La tecnica CT è sviluppata per costruire agenti che operino senza fare ricorso a regole (ad esempio di comportamento economico) definite a priori. Si consideri (fig. 2) lo strato di output di una rete neurale artificiale deputata a determinare le scelte di un agente; lo strato è suddiviso in due parti, caratterizzate come: (1) nodi di uscita relativi ad azioni da compiere (congetture relative a quelle azioni); (2) nodi di uscita relativi agli effetti di tali azioni (sempre congetture, ora relative a quegli effetti). Con i CT, sia i *targets* necessari per l'apprendimento della rete dal lato delle azioni, sia quelli necessari dal lato degli effetti, sono determinati in modo incrociato. I primi sono costruiti in coerenza con gli output della rete concernenti gli effetti (lato 2), allo scopo di sviluppare la capacità di compiere azioni che producano i risultati attesi. I secondi sono costruiti in coerenza con gli output della rete relativi alle congetture di azione (lato 1), con lo scopo di sviluppare la capacità dell'agente di stimare gli effetti delle azioni che sta compiendo<sup>2</sup>. E l'insegnante esterno? Ovviamente sta nelle regole appena descritte, ma si limita a prescrivere la coerenza tra ciò che si fa e gli effetti attesi e viceversa.

Il metodo è completato dalla introduzione di obiettivi esterni (EO) che eventualmente modificano gli output del lato 1 della rete, introducendo così l'esigenza di adattare coerentemente le azioni del lato 2; simmetricamente, sono possibili proposte esterne (EP) di azioni (es. a seguito di eventi inattesi, dell'imitazione, del caso), che modificano gli output del lato 2, cui dovranno adattarsi le congetture sugli effetti del lato 1.

I parametri o pesi della rete si formano durante i cicli di azione e apprendimento, che sono necessariamente pochi, per cui l'azione dell'algoritmo di apprendimento ha in pratica effetto soltanto sui parametri tra gli strati *hidden* ed output, con la rete che riproduce in modo mediato gli ultimi valori generati. Si noti che per quanto sin qui descritto il meccanismo a rete neurale potrebbe essere rimpiazzato da qualsiasi algoritmo che adattasse gradualmente i propri valori ai CT via via determinati.

Quel che però conta è che nella strategia dei CT, al termine di un ciclo di esperimento, ogni agente-rete neurale possa apprendere ripetutamente quanto accaduto, adattando i pesi delle matrici della rete neurale ai dati da essa stessa generati: ciò con riferimento agli input registrati passo a passo ed agli output prodotti, il tutto con un adeguato numero di cicli di applicazione della *backpropagation*. Dopo l'apprendimento di lungo periodo ora descritto diventa possibile utilizzare la rete senza CT, superando pienamente la difficoltà dell'insegnante esterno segnalata sopra.

Resta il problema della plausibilità intrinseca dei CT. Mi farò scudo di Hayek che, quando ritorna (Hayek 1952, V) agli interessi giovanili per la psicologia richiama indirettamente Pope<sup>3</sup>, sostenendo «*A great deal of explanation would be necessary were I to try and justify why an economist ventures to rush in where psychologists fear to tread*». Con quello scudo mi azzardo a richiamare un riferimento di Zimmer (2006, 78) a Sarah-Jayne Blakemore (University College di Londra), secondo la quale il nostro

<sup>2</sup> Per gli aspetti più tecnici vedere Terna (1995, 2000); inoltre si rinvia nuovamente alla sintesi che si trova a <http://web.econ.unito.it/terna/ct-era/ct-era.html>.

<sup>3</sup> Alexander Pope (1711), *An Essay on Criticism*. Lewis, London, verso 625: «*For fools rush in where angels fear to tread.*»

cervello, quando invia un comando per muovere una parte del corpo, trasmette due segnali. Il primo va verso le regioni cerebrali che controllano le parti del corpo da muovere, mentre il secondo si dirige verso le regioni che supervisionano i movimenti. Il nostro cervello si serve di questa copia dell'informazione per prevedere la sensazione che l'azione produrrà. Se la sensazione ricevuta non corrisponde alla nostra previsione, il cervello nota la differenza. La discrepanza ci obbliga a prestare più attenzione a ciò che stiamo facendo, e ci induce a compensare l'azione, ottenendo così il risultato che desideravamo. L'analisi non è lontanissima dall'impostazione dei CT. Sulla previsione che il cervello effettua a proposito della conseguenza delle azioni vedere Blakemore e Frith (2003).

Nell'appendice B, a scopo illustrativo, si trova una applicazione del metodo dei CT al movimento di un agente che cerca un obiettivo. Tutto il materiale relativo ai può essere reperito dalla mia pagina web<sup>4</sup>.

#### COMPLESSITÀ E EMERGENZA

In un capoverso famoso sulle proprietà distintive degli ordini spontanei, Hayek (1973, p.38) sostiene che (i corsivi sono nell'originale):

Un effetto della nostra abituale identificazione dell'ordine con un ordinamento costruito o con un tassonomia, è la nostra tendenza ad estendere a tutti gli ordinamenti proprietà che sono necessariamente possedute solo da disposizioni intenzionali. Si tratta di casi di ordini relativamente semplici, limitati ad un moderato grado di complessità, tanto che il costruttore possa ancora dominare l'insieme; sono casi *concreti*, nel senso che la loro esistenza possa essere intuitivamente percepita con l'osservazione; infine, essendo stati costruiti deliberatamente, *servono* (o sono serviti) *a qualche scopo* del costruttore. Nessuna di queste caratteristiche appartiene necessariamente ad un ordine spontaneo o *cosmo*. Il grado di complessità non è limitato a ciò che la mente umana può padroneggiare. La loro stessa esistenza può non essere evidente ai nostri sensi, ma può derivare da relazioni puramente *astratte* che noi possiamo solo ricostruire mentalmente. Infine, non essendo stato costruito, *non può* legittimamente essere *ascritto ad un scopo*, per quanto la nostra consapevolezza della sua esistenza possa essere estremamente importante per il nostro successo nel raggiungere una grande varietà di differenti intenti.

Gli ordini spontanei non sono necessariamente complessi, ma al contrario degli ordinamenti frutto di azioni intenzionali umane, possono raggiungere qualsiasi grado di complessità. Il nostro assunto centrale è che tali ordinamenti molto complessi, comprendenti più dettagli di quanto qualsiasi cervello possa conoscere o considerare, possano essere causati solo da forze che inducono la formazione degli ordini spontanei.

In termini contemporanei, nel definire il concetto di complessità si sottolineano la quantità in gioco nel fenomeno, le azioni e interazioni delle parti, l'emergenza di fatti

---

<sup>4</sup> <http://web.econ.unito.it/terna/> cercando sub *recent publications* o nel materiale del corso di Dinamica economica (corso ora cessato, ma il materiale è in linea).



imprevisti o imprevedibili (il modello dei camaleonti, introdotto al termine, è interessante anche in questa prospettiva). Ma, sullo sfondo, la lettura hayekiana è estremamente arricchente, con l'indicazione di costruttori e scopi che risultano assenti negli ordinamenti spontanei, propriamente quando molto complessi.

Con Donzelli (1988, p.45):

(. . .) quando la complessità delle circostanze di cui occorre tenere conto trascende le capacità di comprensione e di controllo di ciascun singolo individuo, il coordinamento delle azioni individuali può essere conseguito soltanto attraverso i meccanismi impersonali che governano l'ordine spontaneo. L'ordine spontaneo ci permette quindi di giungere là dove il metodo basato sull'organizzazione fallisce; ma proprio a causa di questa sua maggiore complessità, l'ordine spontaneo risulta molto meno preciso e controllabile di quanto non sia l'ordine generato da una organizzazione.

Ancora qualche precisazione. Il campo della complessità è assai vasto e dal punto di vista di chi scrive il principale strumento per trattare la complessità è la simulazione basata su agenti. Non è però l'unica prospettiva dall'angolo visuale degli strumenti. Una finestra sugli strumenti si trova in Flake (1998), con il codice di calcolo e il materiale illustrativo dei diversi capitoli disponibile all'indirizzo in linea riportato in bibliografia.

Complicato e complesso sembrano sinonimi, ma non nel linguaggio scientifico, almeno da oltre 30 anni, cioè da «*more is different*», motto della complessità tratto dal titolo da un famoso articolo (Anderson, 1972) pubblicato dal futuro premio Nobel per la fisica. Anderson studiava che cosa accade quando degli elementi – atomi, molecole, formiche, persone, automobili ... -, in un certo numero, interagiscono. L'interazione spesso porta ad una disordinata interdipendenza con la difficoltà a comprendere che cosa accade e che cosa «emerge» da quella interazione: ad esempio un gigantesco ingorgo di autoveicoli che, quando finalmente ne usciamo, non lascia scoprire come sia nato. La chiave di volta della complessità sta nel numero delle parti in causa, con i correlati livelli di interazione e di direzione dei legami causali, come nell'esempio dell'acqua che bolle: la transizione che spezza l'ordinato crescere della temperatura in un tazza d'acqua, perché l'acqua diventa vapore e non è più nella tazza, è un fenomeno emergente dall'interazione delle molecole e dipende dal numero delle molecole, tanto che non si produce al di sotto di una certa soglia. Lo stesso accade in economia e più in generale nelle scienze sociali?

Con Anderson (1972, p.396) possiamo affermare di sì, con queste righe conclusive dell'articolo, solo apparentemente scherzose.

In conclusione, propongo due esempi economici di quel che spero di avere detto. Marx ha sostenuto che le differenze quantitative diventano qualitative, ma un dialogo in Parigi negli anni '20 sintetizza la cosa ancor più chiaramente:

FITZGERALD: I ricchi sono differenti da noi.

HEMINGWAY: Sì, hanno più soldi.

## SIMULAZIONE ED EMERGENZA CON AGENTI INTELLIGENTI: IL SORPRENDENTE MONDO DEI CAMALEONTI

Come si chiede Vriend (2002), «*Was Hayek an Ace?*», intendendo un *Agent based Computational Economist*, con il doppio significato di «asso». Molto sullo sfondo, ma certamente sì.

Da Hayek ai camaleonti il passo non è breve, ma non è del tutto insensato: ecco perché proponiamo, ancora ai suoi primi passi, un modello ad agenti basato sulla metafora dei camaleonti, scritto utilizzando una tecnica di simulazione basata su Python ([www.python.it](http://www.python.it)) che ripropone il protocollo guida disegnato dal progetto Swarm ([www.swarm.org](http://www.swarm.org)): si tratta di SLAPP (*Swarm Like Agent Protocol in Python*), reperibile, con alcuni sviluppi del modello dei camaleonti, progressivamente aggiornati, a <http://eco83.econ.unito.it/terna/slapp/>.

Per fare esperimenti con i camaleonti si può scaricare la versione iniziale numerata 1 o quella più avanzata numerata 4 (o con numero maggiore), ma occorre installare Python ed apprendere un minimo della tecnica necessaria per utilizzarlo. Oppure si può utilizzare una traduzione in NetLogo (<http://ccl.northwestern.edu/netlogo/>) della versione 4 semplificata; il risultato è utilizzabile con un normale browser dell'internet, purché si abbia Java installato, con la possibilità di disconnettersi dopo aver visualizzato il programma per eseguirlo fuori linea, in quanto tutto il necessario è in quel momento disponibile (indirizzo <http://eco83.econ.unito.it/terna/chameleons/chameleons.html>).

Nella versione NetLogo (si veda la fig. 3), con *setup* compaiono i camaleonti; con *go* iniziano a muoversi e cambiano colore con la regola che se due camaleonti di colori differenti si incontrano, assumono il terzo colore (un blu e un rosso diventano entrambi verdi ecc.). Se diventassero tutti dello stesso colore (è possibile, provate con *num* basso, ad es. meno di 10, e fate funzionare il programma molto a lungo, anche se il fenomeno potrebbe pure presentarsi dopo pochi cicli), non ci sarebbero altri cambiamenti di colore.

E se i portatori di uno dei colori non volessero cambiare aspetto? Oppure volessero intensamente cambiarlo? Nel prossimo paragrafo è presentato un modello di apprendimento che costruisce le due forme di «mente» da utilizzare per sfuggire agli altri camaleonti (metafora: resistere all'innovazione o al cambiamento) o per inseguire gli altri camaleonti (metafora: diffondere l'innovazione o il cambiamento). Nella versione in linea (si veda ancora la fig. 3), agendo su *smart\_red* o *smart\_green* o *smart\_blue*, tutti con scelta tra «no» oppure «R» oppure «C» (R sta per *runner* e C per *chaser*), si attiva il meccanismo di fuga o di inseguimento, entrambi fondati su reti neurali artificiali che incorporano i frutti di un apprendimento con rinforzo. Provando solo con uno dei colori in condizione R si nota chiaramente l'effetto di difesa, con i camaleonti del colore intelligente che cercano di raggrupparsi e di restare fermi in zone sicure. Anche con due colori in condizione R l'effetto è evidente e ben si notano gli spostamenti difensivi dei colori in stato R quando uno dei (pochi) camaleonti del terzo colore si avvicina; con tre in condizione R, rapidamente tutti trovano una condizione di

prudente difesa e ... non si muovono più, quindi addio innovazione o diffusione di idee ...

Isolando le componenti apparenti del comportamento con mente di tipo R, che paiono essere lo stare fermi e il raggrupparsi, possiamo innanzi tutto provare a azzerare i movimenti di una o due categorie di camaleonti, portando tutto a sinistra il cursore *redMaxStep* o di altro colore. Il vantaggio c'è, temporaneamente anche maggiore di quello derivante dal comportamento *smart*, ma con forti oscillazione e regressi nel numero degli agenti che preservano il proprio colore, mentre nell'azione con mente quel numero si stabilizza dopo essere cresciuto. Ha ancora meno successo la strategia di raggrupparsi con *red\_close\_to\_red* o per altro colore, se è attivo il selettore *not\_to\_close*, perché si tratta di un comportamento difensivo privo di intelligenza nei movimenti; se si disattiva il selettore *not\_to\_close* i camaleonti che scelgono di stare vicini vanno tutti nello stesso punto, rendendo molto difficile individuarli per tutti gli altri (la torre d'avorio dell'isolamento?); non è una strategia – pur vantaggiosa – interessante, in quanto consiste nel nascondersi. Si verifica in questo modo che i meccanismi di difesa o di offesa emersi dalla simulazione non corrispondono a comportamenti *à la* Schelling (rif. Schelling, 1978) fondati su regole semplici, ma derivano da azioni pianificate con regole articolate.

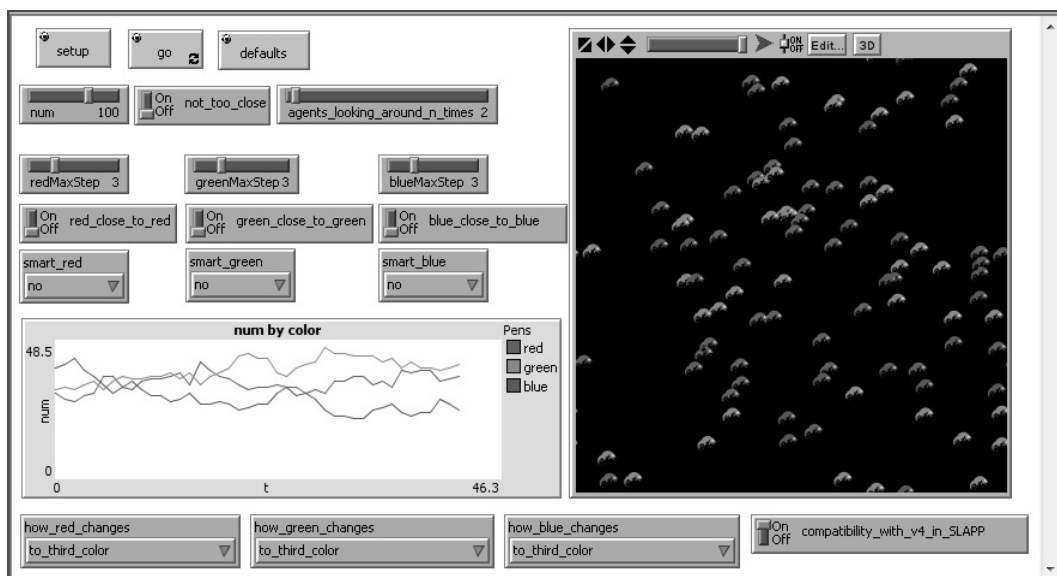


Fig. 3. L'interfaccia della simulazione nella versione in NetLogo.

Ancora sull'interfaccia in NetLogo, un accenno alla variabile che si riferisce all'accuratezza nella ricerca dei vicini, *agents\_looking\_around\_n\_times*, da parte degli agenti che vogliono avvicinarsi ai propri simili (il valore 2 è sufficiente) e al bottone *defaults*, che ristabilisce tutti i valori di partenza, comprese le posizioni dei cursori.

Quello denominato *compatibility\_with\_v4\_in\_SLAPP* è relativo ad una voluta asimmetria, presente nella versione 4 in SLAPP e non utilizzata nella versione NetLogo, del comportamento delle regole di cambiamento gestire dai selettori dell'ultima riga; su questi ultimi torneremo più avanti.

Il modello completo, con i camaleonti inseguitori e innovatori, disponibile in SLAPP, o in NetLogo in linea, permette, pur solo in una prima fase di lettura, interessanti considerazioni.

#### TECNICA DI SVILUPPO DEI COMPORTAMENTI DEI CAMALEONTI

L'esperimento vedrà dunque agenti con e senza «mente» muoversi in un ambiente non strutturato, cioè privo di regole esplicite, salvo quella del cambiamento dei colori. Il mondo è popolato da camaleonti colorati; nella versione più semplice con tre colori, ma ne esiste anche una a cinque colori. I camaleonti vivono in un mondo «a quadretti»: in un quadretto può stare qualsiasi quantità di camaleonti. Il mondo è rappresentato da un quadrato o da un rettangolo, non è limitato dai bordi: se si esce dal lato sinistro si rientra dal destro e nello stesso modo si opera con i bordi superiore e inferiore; tecnicamente si tratta della proiezione piana di un toroide. I camaleonti senza mente si muovono nell'ambiente in modo del tutto casuale, mentre quelli intelligenti si muovono in base alla posizione dei vicini, allontanandosi dal pericolo di cambiare colore se sono conservatori o fuggitivi o avvicinandosi all'opportunità di farlo se sono agenti cambiamento o inseguitori.

La mente dei camaleonti intelligenti è stata realizzata con più reti neurali artificiali che incorporano le informazioni prodotte da un processo di apprendimento con rinforzo (*reinforcement learning*) realizzato secondo una procedura prova ed errore.

Lo scopo dell'apprendimento con rinforzo nel nostro caso (Lin, 1992; Kaelbling e Moore, 1996; Sutton e Barto, 1998) è quello di cercare un criterio di collegamento (*mapping*) ottimale tra lo stato in cui l'agente si trova e l'azione che deve intraprendere. La costruzione del criterio avviene mediante l'assegnazione di ricompense e punizioni, a seconda che l'azione compiuta in presenza di un dato stato abbia sortito o no un effetto positivo. Il tutto, nel nostro caso, è memorizzato dalle reti neurali.

L'algoritmo differisce dall'apprendimento supervisionato normalmente utilizzato per le reti neurali, in quanto le coppie *input-target* non sono necessarie a priori, mentre l'agente è lasciato libero di esplorare l'ambiente. Come con i CT, si cerca in questo modo di superare almeno in parte una delle critiche riportate in Zorzi (2006), quella dell'impossibilità di conoscere a priori i comportamenti degli agenti oggetto dei nostri esperimenti, se lo scopo è quello di far evolvere quei comportamenti. Ovviamente non si supera il problema della plausibilità del metodo rispetto alle modalità di apprendimento naturali, perché qui abbiamo solo sostituito ai *patterns* necessari a priori una regola capace di generarli, sia pure in modo *live*, interagendo con l'ambiente; certo non si tratta di un passo come quello dei CT, in cui è solo più una regola di coerenza interna dell'agente a generare i *patterns*.

Il modello base dell'apprendimento con rinforzo dispone: di un insieme di stati  $S$  relativi all'ambiente o ad una parte di esso; di un insieme di azioni  $A$ ; di un insieme di ricompense rappresentate da numeri reali. Ad ogni istante  $t$  l'agente si trova in uno stato  $s_t$  e può intraprendere un'azione  $a_t$ . Una volta compiuta l'azione, l'agente si trova in un nuovo stato  $s_{t+1}$  e riceve una ricompensa  $r_{t+1}$ . Sulla base di queste interazioni, l'agente intelligente svilupperà un criterio di collegamento tra input (stati) e azioni, secondo le ricompense, per massimizzare la quantità:

$$R = \sum_t \gamma^t r_t \quad (1)$$

definita convenzionalmente utilità; il parametro  $\gamma$  è un fattore di sconto, con valori tra 0 ed 1 e regola l'importanza relativa tra le varie ricompense ottenute nel tempo, attenuando quella delle ricompense troppo lontane (in avanti) nel tempo.

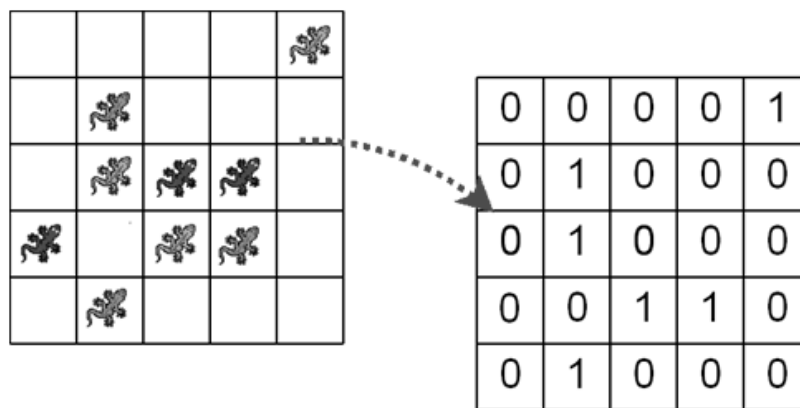


Fig. 4. Il camaleonte al centro della griglia (sarebbe rosso) genera la matrice corrispondente allo stato che osserva<sup>5</sup>. I valori 1 compaiono dove ci sono camaleonti di altro colore.

Esaminiamo ora *stati*, *azioni* e *ricompense*. Il camaleonte intelligente percepisce il proprio stato esaminando l'ambiente intorno a sé, rappresentato dalle 25 celle del quadrato 5x5 centrato sulla sua posizione. Costruisce così una matrice della stessa forma, assegnando valore 1 alle celle occupate da almeno un camaleonte di altro colore e valore 0 nei restanti casi, come in fig. 4. La dimensione dello spazio esaminato (qui 5x5) rappresenta una forma di razionalità limitata. Si potrebbe estendere a casi 7x7 e così via, tenendo però conto della conseguenza sul numero dei possibili stati, che cresce vertiginosamente, essendo nel caso del 5x5 pari a  $2^{24}$ , se non si conteggia la casella centrale (il che non è del tutto giusto). Con lo stesso criterio, 7x7 corrisponde a  $2^{48}$  stati possibili; si tratta di una grandezza dell'ordine di 3 seguito da 14 zero! Già comunque

<sup>5</sup> Sappiamo che piuttosto che camaleonti sembrano gechi ... Abbiamo rimediato nella versione sviluppata in NetLogo.

nel caso del 5x5 una ricerca della soluzione esplorando tutti i possibili stati richiederebbe la gestione di una casistica di quasi 17 milioni di situazioni, di cui certo molte speculari, ma che non necessariamente debbono dare luogo a mosse altrettanto speculari in un sistema di rappresentazione volutamente imperfetta delle scelte, come del resto avviene nella nostra mente.

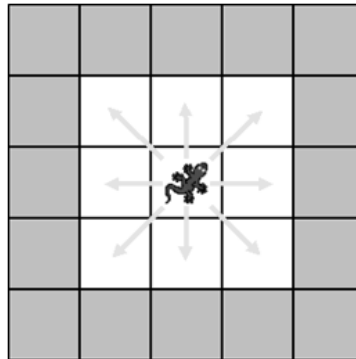


Fig. 5. Il camaleonte può restare fermo o spostarsi in una cella adiacente. Le celle grigie sono fuori portata.

Le azioni che l'agente camaleonte può compiere sono nove e corrispondono allo spostamento di un passo lungo una delle otto direzioni individuabili nel piano, più la possibilità di restare fermo (fig. 5). Ora descriviamo la costruzione delle scelte per le azioni dei camaleonti che non vogliono cambiare colore e sono quindi conservatori o fuggitivi; è facile argomentare la costruzione opposta, per gli agenti di cambiamento o inseguitori, che vogliono mutare colore.

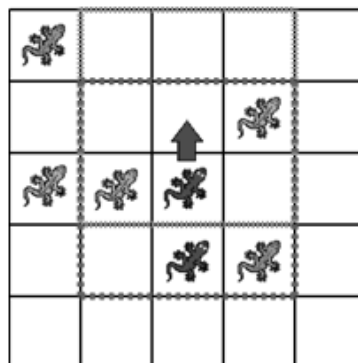


Fig. 6. Spostandosi verso nord, il camaleonte passa da uno stato con  $p = 3$  (riquadro 3x3 centrale) ad uno con  $p = 2$  (riquadro 3x3 superiore), ottenendo una ricompensa non normalizzata pari ad 1.

Per quanto riguarda l'assegnazione delle ricompense e l'apprendimento, la procedura è la seguente.

- Si assegna ad ogni stato un grado di pericolo  $p$  che va da 0 ad 8 a seconda che le 8 caselle adiacenti siano occupate o no da camaleonti di un colore diverso. Il grado di pericolosità dello stato rappresentato nella fig. 4 è pari a 4, poiché tanti sono i camaleonti di altro colore nelle immediate vicinanze.
- Nel passaggio da uno stato all'altro si assegna una ricompensa pari alla differenza dei gradi di pericolo dei due stati (fig. 6). Questo valore è normalizzato rispetto alla massima ricompensa (o alla massima punizione) ottenibile dallo stato di partenza (tab. 1).

|                       |   | $p$ STATO DI ARRIVO |    |    |    |    |    |    |    |   |
|-----------------------|---|---------------------|----|----|----|----|----|----|----|---|
|                       |   | 0                   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8 |
| $p$ STATO DI PARTENZA | 0 | 0                   | -1 | -2 | -3 | -4 | -5 |    |    |   |
|                       | 1 | 1                   | 0  | -1 | -2 | -3 | -4 | -5 |    |   |
|                       | 2 | 2                   | 1  | 0  | -1 | -2 | -3 | -4 | -5 |   |
|                       | 3 | 3                   | 2  | 1  | 0  | -1 | -2 | -3 | -4 |   |
|                       | 4 | 4                   | 3  | 2  | 1  | 0  | -1 | -2 | -3 |   |
|                       | 5 | 5                   | 4  | 3  | 2  | 1  | 0  | -1 | -2 |   |
|                       | 6 |                     | 5  | 4  | 3  | 2  | 1  | 0  | -1 |   |
|                       | 7 |                     |    | 5  | 4  | 3  | 2  | 1  | 0  |   |

Tab. 1. Le caselle in grigio chiaro indicano le massime ricompense ottenibili nel passaggio tra due stati; quelle in grigio medio indicano le massime punizioni. Le celle in grigio più scuro indicano situazioni che non possono verificarsi. I casi limite in cui è presente il valore 0 sono trattati a parte.

- Si calcolata l'utilità, come definita nella (1), relativa alla coppia stato-azione  $(s_t, a_t)$  intervenuta al tempo  $t$  della mossa, come la somma non scontata ( $\gamma = 1$ ) delle ricompense normalizzate ai tempi  $t + 1$  e  $t + 2$ :

$$R_t = \frac{r_{t+1}}{\max(r_{t+1})} + \frac{r_{t+2}}{\max(r_{t+2})} \quad (2)$$

- Nel caso in cui l'azione provochi l'incontro con un camaleonte di colore diverso, l'utilità sarà è posta pari ad un minimo (un massimo, nel caso degli inseguitori). Valori di  $R_t$  superiori ad 1 sono normalizzati ad 1; i valori negativi sono posti a 0.
- La «mente» è costituita da 9 reti neurali (vedere l'appendice A, per una breve presentazione delle reti), una per ognuna delle 9 possibili azioni (fig. 5), tutte aventi 26 nodi di ingresso utilizzati per identificare le 25 celle dello stato, ciascuna con

valore 0 o 1, più un nodo *bias*, e 10 nodi nascosti più il *bias* (5 nodi nascosti più il *bias* si sono rivelati sufficienti nel caso dei camaleonti inseguitori).

Dopo ogni mossa di ciascun camaleonte, il relativo valore  $R_t$  è utilizzato come target (valore  $y$  nell'appendice A) alla rete associata all'azione  $a_t$  intrapresa (abbiamo qui nove possibili azioni e nove possibili reti neurali; dato che l'input-stato è comune a tutte le reti, si sarebbe anche potuto utilizzare un vettore  $y$  di 9 elementi, con un'unica rete neurale, ma la difficoltà di determinazione dei parametri sarebbe stata ben maggiore). Si esegue quindi un passo dell'algoritmo della *backpropagation* con lo stato  $s_t$  come input  $x$ , sempre secondo la convenzione dei simboli dell'appendice A.

Per velocizzare l'apprendimento, questa operazione è stata ogni volta ripetuta in modo proporzionale al valore assoluto di  $R_t$ , in questo caso non normalizzato.

- L'apprendimento è condotto facendo muovere in modo casuale tutti i camaleonti, quindi senza camaleonti intelligenti. Se ogni azione è esplorata un numero sufficiente di volte avremo, ad apprendimento ultimato, una capacità di movimento prossima a quella ottimale. Si tenga però conto del fatto che il numero dei possibili stati è molto elevato e che in un mondo non particolarmente denso molte combinazioni tra quelle con molte caselle delle 25 poste pari a 1 non saranno mai state viste; del resto, altrettanto difficilmente analoghe situazioni si presenteranno nella fase di applicazione. Uno sviluppo del lavoro può consistere nell'introdurre l'apprendimento in presenza di camaleonti intelligenti, sia fuggitivi, sia inseguitori.
- Al momento dell'applicazione, dopo l'apprendimento, la mossa scelta in presenza di un determinato stato è quella la cui rete neurale associata produce l'output più elevato. Nell'appendice C si trova un commento ulteriore all'algoritmo utilizzato.

Dell'applicazione esiste anche una versione più completa, ancora in sviluppo; al momento si tratta della n.4 nel sito indicato, ma ci saranno nuove versioni. Le varianti nella versione in sviluppo – parzialmente incorporate nella trascrizione in NetLogo - sono date: dalla possibilità di scegliere inizialmente numeri di camaleonti diversi per colore e sino a cinque colori; la possibilità di gestire in modo articolato i cambiamenti di colore, compreso il mantenimento del colore corrente in caso di incontro con un agente di colore diverso. In prospettiva è prevista l'introduzione dei CT in alternativa all'apprendimento con rinforzo, per aumentare la plausibilità dello schema di apprendimento.

I cambiamenti di colore nel caso tre colori sono definiti semplicemente, con la dinamica vista dal lato di uno dei due camaleonti, supposto di colore A, mentre l'altro è di colore B. Se  $A = B$  allora non succederà nulla; se A è diverso da B allora i camaleonti cambieranno il loro colore in C. Con più colori è necessario specificare una lista di colori destinazione, appartenente per convenzione a uno dei due camaleonti, quello che nella sequenza degli accadimenti si accorge per primo dell'altro<sup>6</sup>. I colori A e B sono

---

<sup>6</sup> Nello sviluppo dimostrativo in linea con NetLogo i comportamenti, semplificati, sono trattati in modo simmetrico, quale che sia tra due camaleonti a iniziare l'azione del cambiamento di colore. A partita di



esclusi per la scelta di C, a meno che siano esplicitamente inclusi; le possibilità sono: cambiare assumendo il colore più raro in quel momento (compresi A e B, se sono stati inclusi); oppure assumendo il colore più frequente (con A e B, come detto); oppure si può indicare un elenco esplicito di possibili colori, anche uno solo e anche, volendo, il solo A o il solo B. In questo modo possiamo avere camaleonti che non cambiano colore e diffondono il proprio colore. Una variante dello sviluppo in NetLogo è quella di camaleonti che mantengono il proprio colore e consentono il cambiamento verso il terzo colore del camaleonte incontrato, oppure che non cambiano colore e modificano il colore del camaleonte incontrato imponendogli il proprio. Se si opera con un elenco di colori si deve anche scegliere un criterio probabilistico di scelta: equiprobabile; con probabilità decrescente dal più raro al più comune; con probabilità decrescente dal più comune al più raro. Un'ulteriore variante dello sviluppo in NetLogo riguarda i camaleonti inseguitori: nella versione in Python è previsto che, quando nell'intorno non ci sono agenti di altri colori adottano, adottino forzatamente un movimento casuale; in NetLogo continuano a seguire la mente artificiale delle 9 reti neurali, con un effetto di movimento dall'altro verso il basso derivante dai parametri delle reti così come si sono determinati nella fase di apprendimento, applicati ad un vettore di input tutto di zero; in altri cicli di prova ed errore potrebbero risultare andamenti diversi; si noti che il vettore nullo in input è in sé privo di informazioni utili per scegliere una direzione di movimento.

I camaleonti ovviamente potranno muoversi in modo casuale, oppure essere dotati di una mente (le nove reti neurali) da fuggitivi, cioè avversi al cambiamento o conservatori, oppure da inseguitori, propensi al cambiamento o innovatori.

#### CONSEGUENZE DEL COMPORTAMENTO DEI CAMALEONTI

L'interfaccia (fig. 3) della simulazione in NetLogo presenta tre possibili stati per ogni colore rispetto alla mente utilizzata (*smart\_colore*), con *no* = movimento casuale, *R* = movimento da fuggitivo e *C* = movimento da inseguitore, più altre tre possibili scelte per ciascun colore, relative alle variabili *how\_red\_changes*, *how\_green\_changes*, e *how\_blue\_changes* con i possibili valori, per ciascuna delle tre varianti di colore, *to\_third\_color* (comportamento normale), *red\_keeps\_red\_other\_changes* (non cambiano colore, ma l'agente incontrato assume il terzo colore) e *red\_keeps\_red\_other\_red* (non cambiano colore e lo attribuiscono anche all'agente incontrato). Tutte le prove sono svolte con la compatibilità *off* rispetto alla v.4 in SLAPP (questione molto tecnica per confronto con quei risultati).

---

condizioni ciò privilegia il camaleonte secondo dei due; nel caso in cui le regole di cambiamento di colore non siano uguali tra i soggetti e solo un camaleonte disponga di una regola forte, come ad esempio non cambiare il proprio colore e cambiare quello dell'altro camaleonte rendendolo uguale al proprio, tale azione sarà compiuta sia che il camaleonte in questione agisca per primo oppure agisca di rimando nella coppia che si è incontrata.

| N. caso | <i>red</i>                  |   |                          | <i>green</i>                  |  |                          | <i>blue</i>                  |   |                        |
|---------|-----------------------------|---|--------------------------|-------------------------------|--|--------------------------|------------------------------|---|------------------------|
|         | <i>smart_</i><br><i>red</i> | <i>how_</i><br><i>red_</i><br><i>changes</i>  | sintesi<br><i>red</i>    | <i>smart_</i><br><i>green</i> | <i>how_</i><br><i>green_</i><br><i>changes</i> | sintesi<br><i>green</i>  | <i>smart_</i><br><i>blue</i> | <i>how_</i><br><i>blue_</i><br><i>changes</i> | sintesi<br><i>blue</i> |
| 1       | <i>R</i>                    | <i>to_third_</i><br><i>color</i>              | avversi<br>con mente     | <i>No</i>                     | <i>to_third_</i><br><i>color</i>               | acquisitori              | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 2       | <i>R</i>                    | <i>to_third_</i><br><i>color</i>              | avversi<br>con mente     | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 3       | <i>C</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori<br>con mente | <i>No</i>                     | <i>to_third_</i><br><i>color</i>               | acquisitori              | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 4       | <i>C</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori<br>con mente | <i>C</i>                      | <i>to_third_</i><br><i>color</i>               | acquisitori<br>con mente | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 5       | <i>No</i>                   | <i>red_keeps_red_</i><br><i>other_changes</i> | mediatori                | <i>No</i>                     | <i>to_third_</i><br><i>color</i>               | acquisitori              | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 6       | <i>C</i>                    | <i>red_keeps_red_</i><br><i>other_changes</i> | mediatori<br>con mente   | <i>No</i>                     | <i>to_third_</i><br><i>color</i>               | acquisitori              | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 7       | <i>No</i>                   | <i>red_keeps_red_</i><br><i>other_red</i>     | autoritari               | <i>No</i>                     | <i>to_third_</i><br><i>color</i>               | acquisitori              | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 8       | <i>C</i>                    | <i>red_keeps_red_</i><br><i>other_red</i>     | autoritari<br>con mente  | <i>No</i>                     | <i>to_third_</i><br><i>color</i>               | acquisitori              | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 9       | <i>No</i>                   | <i>red_keeps_red_</i><br><i>other_changes</i> | mediatori                | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>R</i>                     | <i>to_third_</i><br><i>color</i>              | avversi con<br>mente   |
| 10      | <i>C</i>                    | <i>red_keeps_red_</i><br><i>other_changes</i> | mediatori<br>con mente   | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>R</i>                     | <i>to_third_</i><br><i>color</i>              | avversi con<br>mente   |
| 11      | <i>C</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori<br>con mente | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>R</i>                     | <i>to_third_</i><br><i>color</i>              | avversi con<br>mente   |
| 12      | <i>C</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori<br>con mente | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 13      | <i>No</i>                   | <i>red_keeps_red_</i><br><i>other_canges</i>  | mediatori                | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 14      | <i>C</i>                    | <i>red_keeps_red_</i><br><i>other_changes</i> | mediatori<br>con mente   | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 15      | <i>No</i>                   | <i>red_keeps_red_</i><br><i>other_red</i>     | autoritari               | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |
| 16      | <i>C</i>                    | <i>red_keeps_red_</i><br><i>other_red</i>     | autoritari<br>con mente  | <i>R</i>                      | <i>to_third_</i><br><i>color</i>               | avversi con<br>mente     | <i>No</i>                    | <i>to_third_</i><br><i>color</i>              | acquisitori            |

Tab. 2. La sequenza degli esperimenti.

La tab. 2 elenca un piano di esperimenti che può essere agevolmente replicato utilizzando la versione in linea, che però non salva il file dei risultati intesi come numero di camaleonti di ciascun colore in ogni ciclo.

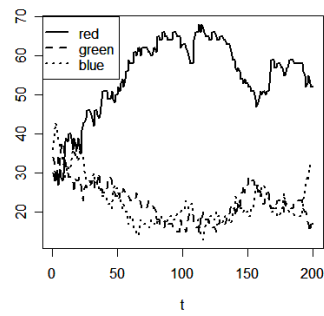
Per fare ciò occorre scaricare il programma dalla stessa pagine in cui funziona e installare NetLogo versione 3.1.4 (o successiva, precedente alla versione 4, che presenterà delle incompatibilità). I lettori sono invitati non solo a replicare gli esperimenti svolti, ma anche a sviluppare nuovi casi. I diversi camaleonti sono portatori di una caratteristica, cioè il proprio colore (elemento tecnologico o idea che sia), ma non tutti vogliono diffonderlo.

Definiamo ora i loro comportamenti, con alcune denominazioni evocative.

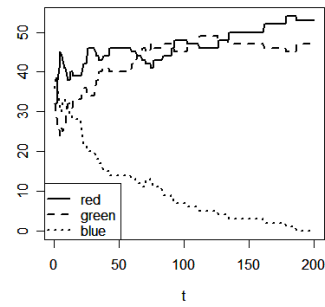
- a) «Diffusori autoritari» o «autoritari» gli agenti che si muovono in modo casuale non cambiando mai il proprio colore e rendendo uguale al proprio quello degli agenti incontrati.
- b) «Diffusori autoritari con mente» o «autoritari con mente» quelli che agiscono come i precedenti e che si muovono utilizzando la mente da inseguitori.
- c) «Diffusori mediatori» o «mediatori» gli agenti che si muovono in modo casuale non cambiando mai il proprio colore, ma fanno cambiare quello degli agenti incontrati, che assumono il terzo colore.
- d) «Diffusori mediatori con mente» o «mediatori con mente» quelli che agiscono come i precedenti e che si muovono utilizzando la mente da inseguitori.
- e) «Diffusori che acquisiscono» o «acquisitori» gli agenti che si muovono in modo casuale cambiando il proprio colore e quello degli agenti incontrati, entrambi mutati al terzo colore.
- f) «Diffusori che acquisiscono con mente» o «acquisitori con mente» quelli che agiscono come i precedenti e che si muovono utilizzando la mente da inseguitori.
- g) «Avversi al cambiamento con mente» o «avversi con mente» gli agenti che si muovono utilizzando la mente da fuggitivi, con lo scopo di evitare contatti con altri agenti se il contatto avviene entrambi sono mutati al terzo colore, a meno che il secondo abbia una regola di tipo (a) o (b) nel qual caso impone il proprio colore (non nel caso in cui sia  $O_n$  la compatibilità con la v.4 in SLAPP).

La tab. 2 riporta lo stato delle scelte con l'interfaccia NetLogo e l'indicazione evocativa del tipo di agenti.

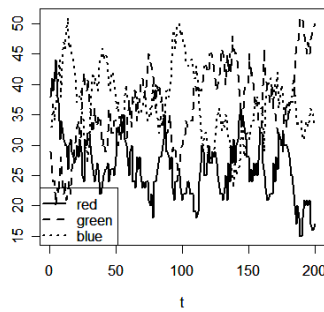
Come si può rilevare dalle figg. 7 e 8, che riportano gli andamenti del numero di camaleonti per colore nei 16 casi della tab.2, esiste una grande varietà di andamenti, che si cogliere a colpo d'occhio osservando i vari grafici.



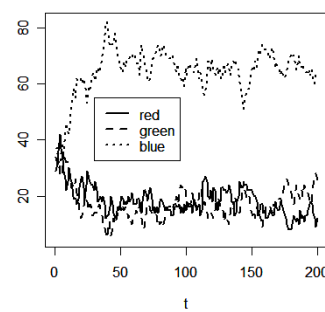
(caso 1)



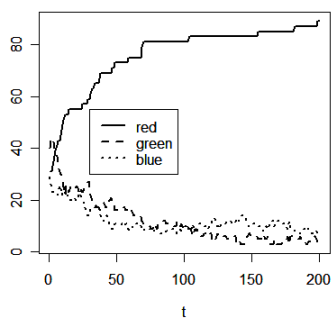
(caso 2)



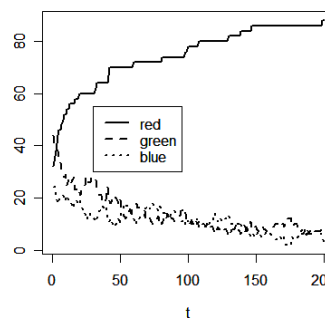
(caso 3)



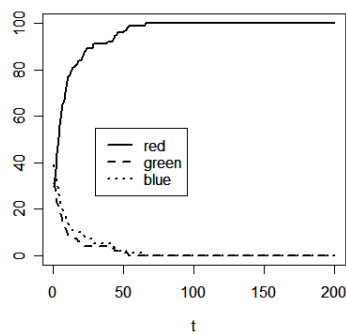
(caso 4)



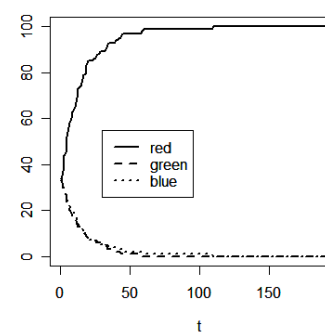
(caso 5)



(caso 6)

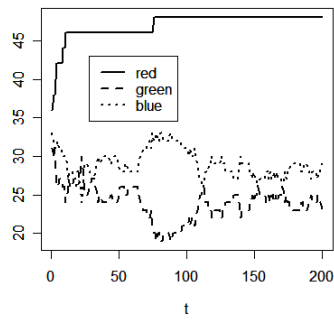


(caso 7)

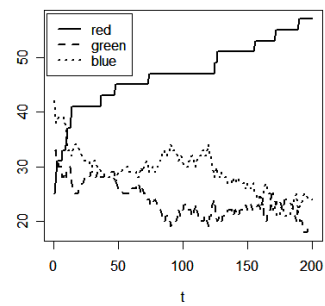


(caso 8)

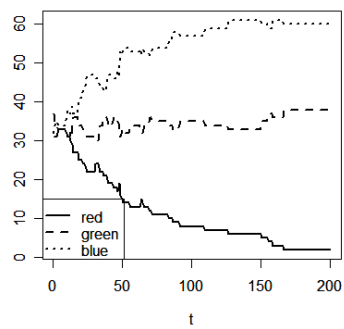
Fig. 7. Andamento del numero dei camaleonti nei casi da 1 a 8.



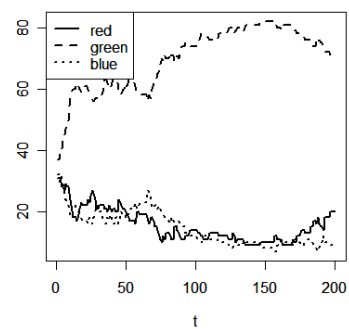
(caso 9)



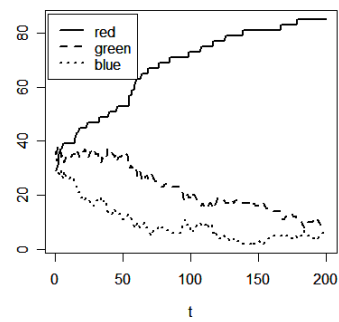
(caso 10)



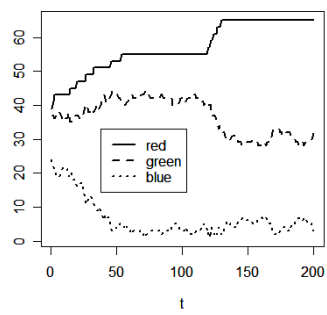
(caso 11)



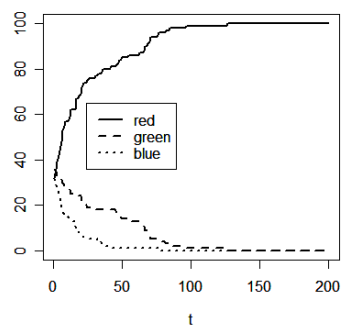
(caso 12)



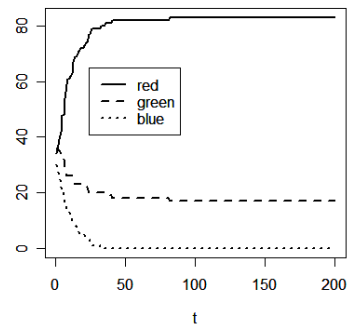
(caso 13)



(caso 14)



(caso 15)



(caso 16)

Fig. 8. Andamento del numero dei camaleonti nei casi da 9 a 16.

L'interpretazione che segue è una chiave di lettura per ragionare sui fenomeni emergenti nel nostro modello di simulazione, evidentemente con una componente di soggettività da parte di chi scrive; questa è del resto una delle caratteristiche centrali di questo tipo di modelli, ovverosia l'essere di ausilio al ragionamento, tanto che uno dei risultati importanti è quello della generazione di nuove idee in chi interpreta il modello via via che ne vede comparire sullo schermo i risultati; si tratta di una esperienza che può essere ripetuta dal lettore. E' l'interpretazione del computer come amplificatore delle capacità di ragionamento.

Il significato della domanda «che cosa avrebbe potuto fare Darwin se avesse avuto un computer a sua disposizione?» in Dawkins (1999) è molto simile alla risposta di Arthur nel dialogo con Horgan che dubita della portata della teoria della complessità (Horgan, 1995); nel dialogo, non è da sottovalutare il contro esempio:

«Se Darwin avesse avuto un computer sulla scrivania» esclama l'economista del Santa Fe Institute W. Brian Arthur «chissà che cosa avrebbe potuto scoprire!». Che cosa, infatti: Charles Darwin avrebbe potuto scoprire una gran quantità di cose sui computer e molto poco sulla natura.

Passiamo ai nostri casi, utilizzando la terminologia definita, richiamata anche con l'indicazione della lettera relativa. Al centro sta una interpretazione su come le idee si diffondono, per contagio o per collaborazione e scambio, consapevole o no. Nel primo caso l'agente che diffonde le idee non modifica il proprio stato per effetto dei contatti con altri agenti, come nella diffusione di un virus; nel secondo caso sia l'agente che propaga le idee sia quello che le riceve si modificano. Si pensi ad esempio ad uno studioso che collabora con studiosi di altre scuole oppure ad un ricercatore accademico che collabora con ricercatori di un centro privato per sviluppare applicazioni: tutti si co-influenzeranno e co-informeranno. I nostri agenti (a) e (b) sono del primo tipo; quelli (c) (d) (e) ed (f), del secondo.

Nel caso 1, con i camaleonti rossi avversi con mente (g) e gli altri due colori operanti come diffusori acquisitori (e), il mondo si popola quasi solo di rossi, che non hanno nulla da diffondere, ma non cambiano mentre gli altri cambiano, producendo anche dei rossi che si avvantaggiano nettamente e, quando in gran numero, si bloccano: un disastro per l'innovazione. Nel caso 2, con anche i verdi avversi con mente (g), le cose non migliorano, in quanto il modo risulta popolato quasi solo di rossi e verdi che, raggiunto il risultato, non si muovono più. Dunque un mondo di avversi con mente e di acquisitori senza strategia, come dire poco attrezzati, produce una situazione di stallo, senza innovazione.

Nel caso 3 i rossi sono portatori di innovazione acquisitori con mente (f) e gli altri sono acquisitori (e): ne deriva una grande variabilità creativa. Non bisogna però esagerare, in quanto nel caso 4, dove anche i verdi sono diffusori acquisitori con mente (f), rossi e verdi si elidono e crescono solo i blu, senza interlocutori.

Nel caso 5 dove compaiono, come rossi, i diffusori mediatori (c) mentre gli altri sono acquisitori (e), si ripresenta una situazione di predominio dei rossi, a scapito della varietà e della diffusione delle idee dei verdi e dei blu. Altrettanto accade nel caso 6, dove i diffusori mediatori sono dotati di mente (d).

Le cose peggiorano ulteriormente nei casi 7 e 8, dove i rossi sono autoritari (a) o autoritari con mente (b) e gli altri acquisitori (e): compare il «pensiero unico».

In presenza di propagatori forti come i mediatori (c), per noi i rossi del caso 9, la presenza di verdi e blu avversi con mente (g) determina una sopravvivenza della varietà, che si attenua però nel caso 10, dove i rossi sono mediatori con mente (d).

Gli acquisitori con mente (f), efficaci nel caso 3, perdono validità nel caso 11, in presenza di verdi e blu avversi (g): spariscono i rossi, restando solo verdi e blu in un mondo che si immobilizza. Nel caso 12, analogo all'11, ma con i blu acquisitori (c), i conservatori avversi al cambiamento - verdi di tipo (g) - stravincono. Quindi il risultato del caso 3 è molto fragile in presenza di comportamenti avversi al cambiamento.

La presenza di rossi mediatori (c) nel caso 13 o mediatori con mente (d) nel caso 14, che restava un po' controbilanciata nei casi 9 e 10, ora con verdi avversi (g) e blu acquisitori (e) porta alla sparizione della componente blu.

Infine i casi 15 e 16, dove ricompaiono gli autoritari (a) o gli autoritari con mente (b), ora con verdi avversi (g) e blu acquisitori (e), confermano la mancanza di varietà e la presenza dominante dei rossi vista nei casi 7 e 8, sia pure con una difesa debole dei verdi conservatori (non utile all'innovazione) nel caso 15 e un po' più forte (ma sempre non utile all'innovazione) nel caso 16; il che potrebbe anche voler dire che dal punto di vista degli autoritari, meglio non avere mente che averla!

Il commento che ne deriva è che i comportamenti forti non facilitano l'innovazione; per la diffusione delle idee con varietà la ricetta vincente è, da un lato, avere diffusori acquisitori dotati di comportamento con mente da inseguitori, ma dall'altro avere interlocutori acquisitori, aperti all'innovazione e allo scambio. Gli altri casi sono molto critici, attribuendo molta forza a chi conserva la situazione presente (gli avversi con mente), oppure con la prevalenza di comportamenti autoritari, molto negativi come si è rilevato. Con Page (2007), non si può che ricordare il potere della differenza e della varietà.

Non so al momento ricavarne una interpretazione, ma una «scena» non ovvia interpretata dai camaleonti, e che si può riprodurre facilmente in linea, è quella che si ottiene scegliendo tutti valori *defaults*, poi indicando C (inseguitori) per *smart\_red*, *smart\_green* e *smart\_blue*; dopo un centinaio di cicli, quando esisteranno solo più pochi raggruppamenti di inseguitori riuniti tra loro e tutti in movimento sullo schermo, indicare *On* per *red\_close\_to\_red* (automaticamente *smart\_red* si pone in stato *no*); quando i camaleonti rossi saranno in un unico punto, indicare *On* per *not\_too\_close* e ... osservare l'enorme «zuffa» che ne deriva.

## RETORICA DEI MODELLI DI SIMULAZIONE E DELLA COMPLESSITÀ

Un accenno finale a come si possano meglio rappresentare i modelli di simulazione, in particolare se fondati su agenti, per comunicarne contenuto, utilizzazione e risultati. Il tema è molto prossimo a quello della rappresentazione dei modelli della complessità, che molto spesso si servono di quel tipo di simulazione. Per il lavoro di ricerca è fondamentale una rappresentazione formale della complessità e dei modelli, in particolare utilizzando l'analisi delle reti tramite i grafi, con una completa presentazione, che è quasi un tutorial, in Christensen e Reka (2006).

Per comunicare i risultati all'esterno del circolo della ricerca, preoccupandoci quindi della retorica dei nostri modelli, dobbiamo invece decidere, con la felice espressione di Roberto Serra, studioso dell'intelligenza artificiale e dell'organizzazione, dove ci collochiamo su un continuo che va dall'astrattezza (i *vetri di spin*, dei fisici) alla divulgazione spinta, con i videogiochi. Personalmente propendo per i videogiochi.

In Bainbridge (2007), articolo che ha conquistato la copertina di *Science*, si affronta il problema, in qualche modo reciproco, dell'utilizzazione di ambienti in linea come *Second Life* o *World of Warcraft*, popolati di agenti che sono avatar (pupazzetti grafici) che rappresentano persone reali che li guidano, con l'aggiunta di agenti comandati da programmi, per compiere esperimenti sulle persone. Oppure, aggiungiamo, gli agenti potrebbero essere solo artificiali, per poter rappresentare i modelli, ad esempio i camaleonti, in forma di videogioco molto sofisticato.

Il campo è apertissimo: sono passati ora dieci anni dal lancio di *Ultima Online*, il titolo della Origin che avrebbe dato il via al boom dei MMOG (*Massive Multiplayer Online Games*), acrostico che ormai compare regolarmente anche su quotidiani e riviste. Il successo commerciale di ambienti come *World Of Warcraft* e *Second Life* ha destato l'interesse di molti, con partecipanti che si contano in unità di milioni.

Ovviamente uno spazio artificiale, ma verosimile, in cui le persone possono interagire in modalità e situazioni analoghe a quelle reali, è un oggetto di grande interesse per la ricerca scientifica in campo sociale, economico, dei comportamenti e delle neuroscienze. Per una discussione e molti esempi applicativi in questa prospettiva, si veda Miglino e al. (2007).

Un mondo virtuale è un ambiente simulato al computer, che ospita contemporaneamente più utenti collegati in rete. Gli utenti interagiscono tra loro e con il mondo tramite, come detto, gli *avatar*, cioè rappresentazioni virtuali più o meno personalizzabili, solitamente di forma umanoide. La plausibilità degli ambienti artificiali è garantita trasferendo al computer, sotto forma di codici informatici, le regole del mondo reale. Particolare attenzione è destinata alla riproduzione dei movimenti, della topografia, della leggi fisiche più rilevanti, in parte anche dell'economia.

L'etichetta «mondo virtuale» è spesso utilizzata come sinonimo di MMOG, anche se, come nel caso di *Second Life*, parlare di gioco non è del tutto appropriato. Esistono



diversi tipi di MMOG, da quelli orientati al gioco, solitamente di ambientazione fantastica, a quelli sviluppati per scopi di carattere didattico ed educativo.

Hanno, comunque, alcune proprietà in comune :

- spazio condiviso: il mondo permette a più partecipanti in rete di incontrarsi, tanto che si registrano riunioni e convegni in *Second Life*;
- interfaccia grafica: il mondo, gli abitanti e gli oggetti che lo popolano, sono rappresentati graficamente, generalmente in tre dimensioni per rendere l'esperienza più coinvolgente; qualcuno ricorderà invece realizzazioni anticipatorie come *Little Italy* del prof. Gianni Degli Antoni, solo testuale, ma certo coinvolgente;
- immediatezza: le interazioni tra gli agenti, siano essi gestiti da un utente o dal calcolatore, hanno luogo in tempo reale;
- interattività: il mondo permette agli utenti di alterare, sviluppare, costruire e inserire nel mondo propri contenuti;
- persistenza: il mondo continua ad esistere, agire ed eventualmente ad evolvere indipendentemente dal fatto che l'utente sia o meno connesso.
- socializzazione: il mondo permette e incoraggia la formazione di gruppi sociali al suo interno, come squadre, gilde, club, condomini, villaggi, ...

Il fascino dei mondi digitali risiede nell'insieme di questi attributi, così come le ragioni di un loro possibile impiego in veste di *laboratori virtuali*. Sebbene i primi passi in questa direzione siano stati compiuti nei decenni passati, le opportunità fornite dai MMOG di ultima generazione rappresentano una svolta. Alcuni ricercatori stanno già lavorando sui MMOG commerciali, realizzando esperimenti di varia natura, utilizzando metodologie diverse a seconda del mondo scelto. *Second Life* risulta, ad esempio, essere molto adatto per la conduzione di esperimenti formali nel campo della psicologia sociale e delle scienze cognitive. Questo perché i ricercatori possono letteralmente costruire attrezzature e strutture simili a quelle reali, per poi reclutare i soggetti del test tra gli abitanti. *World of Warcraft*, d'altra parte, si presta meglio per eseguire statistiche non intrusive sulle reti sociali ed i sistemi economici, in quanto genera automaticamente una moltitudine di dati sulle interazioni sociali ed economiche che avvengono tra i giocatori.

I vantaggi derivanti dall'impiego di questi laboratori artificiali sono molteplici. In primo luogo è possibile svolgere esperimenti difficilmente realizzabili nel mondo reale. Si pensi, per esempio, allo studio delle conseguenze socioeconomiche dell'applicazione di diverse politiche ambientali, oppure alla verifica delle procedure per i piani comunali di protezione civile. Sperimentazioni sono state già effettuate per studiare le reazioni di una popolazione nel caso di diffusione di un'epidemia e sulle modalità del contagio. I soggetti possono essere scelti varcando i confini socioculturali, anche traendoli da gruppi normalmente sottorappresentati; analogamente, gli esperimenti possono essere condotti da studiosi appartenenti a centri di ricerca diversi. La persistenza dei mondi

virtuali, inoltre, permette l'analisi di processi che hanno luogo in tempi lunghi, senza che questo faccia lievitare i costi dell'esperimento.

Il passo successivo è quello di utilizzare gli stessi ambienti per rappresentare simulazioni ad agenti verso chi sta in quel mondo e può anche interagire con la simulazione, o anche solo per presentare il risultato, come un filmato, ad un osservatore esterno.

Molto interessante in questa prospettiva il progetto *Multiverse*, [www.multiverse.net](http://www.multiverse.net), che mette a disposizione degli sviluppatori – gratuitamente, se per usi non commerciali – una base evoluta, programmabile in Java o in Python, per la creazione di mondi virtuali.

Strumenti informatici più semplici che permettono di ottenere risultati prossimi ai videogiochi sono StarLogo TNG, <http://education.mit.edu/starlogo-tng/>, Logo parallelo multi agente come NetLogo, in cui TNG sta per *The Next Generation*. Prossima generazione, con due caratteristiche innovative: si programma muovendo e incastrando blocchi come se si costruisse un puzzle (in questo StarLogo TNG è molto simile ad un altro ambiente, assai utile per imparare – per i giovani e i meno giovani – a programmare; si tratta, sempre con sorgente del codice presso l'MIT, di Scratch, <http://weblogs.media.mit.edu/llk/scratch/>); i risultati delle simulazioni, come si può vedere in linea a <http://education.mit.edu/starlogo-tng/complexity/>, hanno un aspetto molto simile a quello di un videogioco. Oppure abbiamo nuovi ambienti con grafica programmabile, come Flex ([www.adobe.com/it/products/flex/](http://www.adobe.com/it/products/flex/)) o il «sempre verde», e sempre difficile, SmallTalk ([www.smalltalk.org](http://www.smalltalk.org)), ora utilizzabile ad esempio con Squeak ([www.squeak.org](http://www.squeak.org)).

E se le simulazioni non fossero videogiochi? Secondo Bostrom (2003) la probabilità che noi (noi che scriviamo, voi che leggete, la rivista cui appartengono queste pagine, ...) esistiamo solo in una simulazione computerizzata, à la Matrix per intenderci, è elevata. Leggere per giudicare ...

#### SVILUPPI FUTURI

L'uso delle reti neurali artificiali per la costruzione di agenti, e in particolare del relativo gestore e produttore di regole, come indicato nella fig. 1, rappresenta certamente un campo di ricerca da sviluppare, con o senza l'impiego della tecnologia CT.

Occorre poter lavorare in questa direzione con modalità di scrittura del codice informatico molto semplici, per consentire la costruzione dei modelli e la loro verifica anche a chi non è uno specialista dei linguaggi informatici. Per questo è in corso di sviluppo la costruzione del simulatore semplificato in Python, denominato SLAPP. SLAPP riproduce il protocollo di Swarm, cioè una struttura che prevede: un osservatore che crea il modello e gestisce l'output dei risultati, in particolare grafici, tabelle e file; un modello, che crea gli agenti e le loro regole, sotto forma di gestori o generatori di regole (sempre fig. 1). Osservatore e modello operano secondo una sequenza temporale di azioni che, rispettando il protocollo di Swarm, è definita tramite

gruppi di azioni da eseguire secondo una scansione temporale, in modo ripetuto ad ogni ciclo dell'orologio della simulazione. Il tutto è costruito appoggiandosi direttamente su Python, senza necessità di nuove biblioteche di funzioni.

Con SLAPP è in corso di sviluppo la versione CT dei camaleonti, avendo in input lo stato, come sopra con l'apprendimento con rinforzo, e come azione la mossa da compiere; l'effetto è quello conseguente in termini di numero di vicini pericolosi, con l'obiettivo esterno di ridurli. Le congetture possono essere errate, ma i *targets* incrociati generati dai CT indicheranno il numero esatto di vicini pericolosi, con cui effettuare l'apprendimento ripetuto.

Due gli obiettivi: superare quasi del tutto la critica della non plausibilità dell'esigenza di informazioni esterne per generare l'apprendimento degli agenti guidati da reti neurali; scoprire nuove azioni dei camaleonti, sempre virtuali, ma speriamo multicolori.

## APPENDICI

### A. CENNI SULLA STRUTTURA DELLE RETI NEURALI ARTIFICIALI

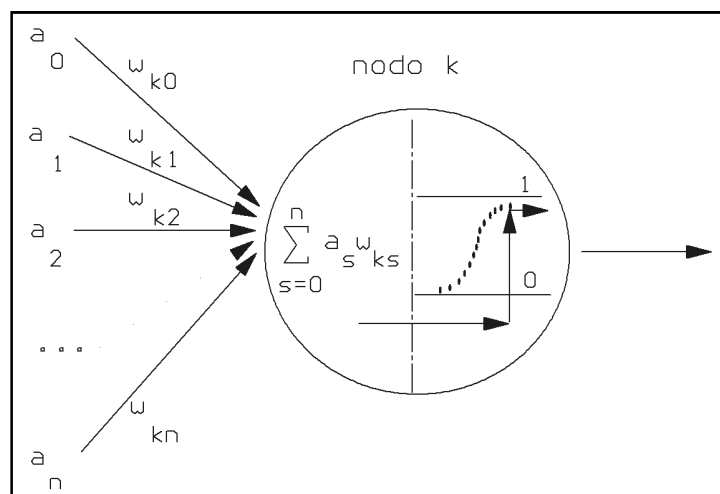


Fig. 9. Uno schema di nodo o neurone artificiale.

Rinviando a Terna (1995) e a Beltratti *e al.* (1996) per un adeguato apprendimento, in fig. 9 è riportato lo schema di un neurone artificiale: più attivazioni o segnali di input, da  $a_0$  ad  $a_n$ , pervengono al neurone  $k$  (dall'esterno; da altri neuroni); le attivazioni sono usualmente confinate nell'intervallo  $0 \div 1$ . Le attivazioni sono moltiplicate (pesate) con valori  $w$ . Nel neurone si sommano i prodotti  $a_j w_{kj}$ ; il risultato è trasformato con una funzione sigmoideale, per lo più la logistica, che confina il risultato in un intervallo dato,

anche operando in un modo assimilabile a quello di una funzione a soglia qual è quella che nei neuroni naturali consente o no l'output nell'assone a seconda degli input raccolti.

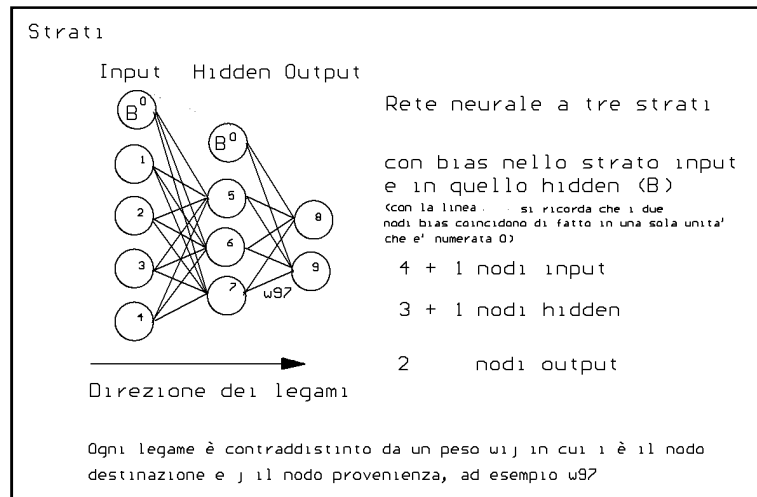


Fig. 10. Una rete di nodi o neuroni artificiale.

La fig. 10 introduce un esempio di rete composta da neuroni artificiali. I nodi di input, qui numerati da 1 a 4, raccolgono le singole informazioni da inviare a tutti i nodi dello strato *hidden*, qui contraddistinti con gli indici da 5 a 7; questi sommano gli input ricevuti, ponderandoli opportunamente, quindi li trasformano tramite una funzione sigmoideale e inviano il risultato a tutti i nodi dello strato output, indicati nel nostro esempio con 8 e 9; questi ultimi nuovamente sommano gli input ricevuti, ponderandoli, quindi li trasformano tramite una funzione come sopra e li emettono quale risultato (con una eventuale ulteriore trasformazione lineare).

La rete deve produrre in uscita un valore predefinito per ogni set di dati in input; i valori di output atteso sono contenuti nell'insieme di dati su cui si fonda l'apprendimento (*training set*); la rete stessa, dopo l'apprendimento con cui si definiscono i pesi  $w$ , sarà verificata su un insieme di dati di controllo (*validation set*) e quindi applicata a dati ulteriori. Gli output attesi, su cui la rete apprende, o quelli contenuti nel *validation set*, sono anche indicati con il nome di *targets*.

Consideriamo i parametro o pesi tra strato input e hidden e tra strato hidden e output della fig. 10 come elementi delle matrici che seguono:

$$\mathbf{A} = \begin{pmatrix} w50 & w60 & w70 \\ w51 & w61 & w71 \\ w52 & w62 & w72 \\ w53 & w63 & w73 \\ w54 & w64 & w74 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} w80 & w90 \\ w85 & w95 \\ w86 & w96 \\ w87 & w97 \end{pmatrix}$$

Trascurando i valori costanti 1 derivanti dai nodi indicati come *bias* (il cui specifico coefficiente  $w$  trasla sull'asse  $x$  il punto di flesso della funzione sigmoideale utilizzata), si ottiene la rappresentazione in forma compatta:

$$y = f(B f(A x))$$

In altri termini una funzione vettoriale di vettori, i cui parametri delle matrici  $A$  e  $B$  sono determinabili con la regressione multipla non lineare oppure, in forma semplificata, con l'algoritmo detto di *backpropagation* dell'errore. In ogni caso, sulla base di *set* di dati ai quali adattare la nostra funzione.

Un brevissimo accenno al funzionamento della *backpropagation*: dato un vettore  $x$  di input la rete, che inizialmente è costruita con pesi  $w$  in  $A$  e  $B$  casuali, produce un risultato  $\hat{y}$  certamente diverso dal valore esatto  $y$ . Si calcolano l'errore  $E = \sum(\hat{y} - y)^2$  e le derivate di  $E$  rispetto a ciascuno dei parametri  $w$ , al fine di correggere i valori  $w$  stessi nella direzione dell'opposto di quelle derivate; ciò allo scopo di minimizzare  $E$ . Il calcolo, per piccole correzioni dei parametri  $w$ , è ripetuto numerose volte per ciascuno dei vettori  $x$  e  $y$  disponibili, alternandoli. In alternativa una funzione a rete neurale, dato un insieme di vettori  $x$  e dei corrispondenti vettori  $y$ , può essere trattata con tecniche di regressione multipla non lineare, con un forte carico computazionale (White, 1989).

## B. APPLICAZIONI DEI CT AL MOVIMENTO

Si introduce a titolo di esempio un esperimento sul movimento di agenti che cercano il cibo, presentato dettagliatamente in Terna (1995). Su un piano con coordinate  $(x,y)$ , il soggetto è inizialmente in  $(10, 10)$ , con il cibo in  $(0, 0)$ . La posizione del cibo è fissa. La rete neurale che simula il soggetto ha in input: la posizione nella direzione  $x$  al tempo  $t - 1$ ; la posizione nella direzione  $y$  al tempo  $t - 1$ ; il movimento nella direzione  $x$ , al tempo  $t - 1$ , con limite  $\pm 1$ ; il movimento nella direzione  $y$ , al tempo  $t - 1$ , con limite  $\pm 1$ . Le coordinate di posizione possono anche essere interpretate come distanza, misurata in coordinate rettangolari, del soggetto dal cibo.

Utilizzando la terminologia dei CT, la rete produce come output due congetture relative agli effetti e due relative alle azioni. Le congetture relative agli effetti sono  $X(t)$  e  $Y(t)$ ,

cioè la posizione al tempo  $t$ . Le congetture relative agli effetti sono:  $\Delta X(t)$ ,  $\Delta Y(t)$ , con tutti i simboli di output che hanno lo stesso significato degli input. I *target*, o valori attesi per l'output, sono:  $X'(t) = X(t-1) + \Delta X(t)$  e  $Y'(t) = Y(t-1) + \Delta Y(t)$ ;  $\Delta X'(t) = \Delta X(t) + X(t) - X'(t)$ , che è l'azione corretta per ottenere  $X(t)$ , incrementando  $\Delta X(t)$  se  $X(t) - X'(t) > 0$  e diminuendolo nel caso opposto;  $\Delta Y'(t) = \Delta Y(t) + Y(t) - Y'(t)$ , come sopra. Per costruzione, essendo il cibo nell'origine e producendo la rete senza apprendimento valori intermedi tra il minimo e il massimo del dominio di valori definito, qui fissato in -2 e 12, accade che la rete ricerca una posizione di coerenza tra congetture di azione e di effetti che fa muovere l'agente nella giusta direzione, come nella fig. 11a. Questo effetto sparisce se l'intervallo sopra indicato produce un valore intermedio esterno: ad esempio con -2 e 50 il movimento sarebbe in direzione opposta.

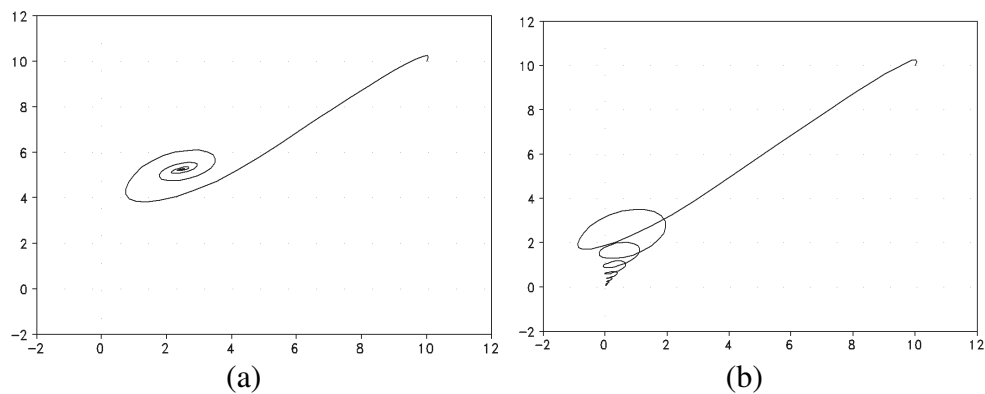


Fig. 11. Esperimento CT di movimento verso (0,0).

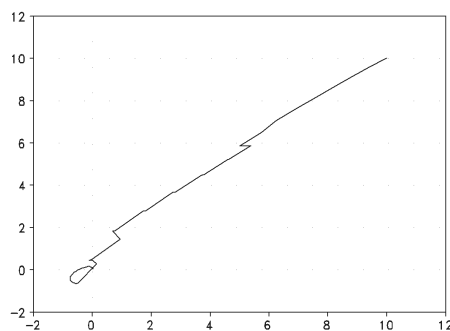


Fig. 12. Esperimento CT di movimento verso (0,0), dopo l'apprendimento a lungo termine.

Il risultato ottenuto per costruzione è comunque superato introducendo un EO (obiettivo esterno) dal lato degli effetti; possiamo ad esempio fare in modo che il soggetto operi per ridurre la distanza dal cibo, ad esempio al 75% (obiettivo non necessariamente raggiungibile) di quella del passo precedente. Si noti che comunque non si danno indicazioni sul movimento da compiere, ma solo sul risultato. L'effetto è quello della

fig. 11b. Nella fig. 12 il caso presentato è lo stesso della fig. 11b, ma con l'effetto di un prolungato apprendimento ripetuto (in questo caso svolto alla fine del periodo).

L'agente simulato raggiunge il cibo con qualche realistica incertezza, muovendosi direttamente nella giusta direzione, senza più far uso dei CT: ora la rete neurale di fatto decide il movimento in funzione dell'input di posizione e di movimento del passo precedente e si tratta di un risultato emergente, non previsto a priori.

### C. COMMENTO ALL'ALGORITMO DI APPRENDIMENTO CON RINFORZO APPLICATO AI CAMALEONTI

Per la realizzazione della simulazione ad agenti con l'apprendimento con rinforzo e reti neurali, si è preferito separare nettamente l'apprendimento dall'applicazione. Si crea quindi prima la mente, facendo muovere a caso i camaleonti, e poi la si utilizza in un nuovo esperimento, dove i camaleonti intelligenti seguiranno il criterio di mappatura memorizzato. Nonostante la razionalità di questi agenti sia molto limitata, l'assegnazione delle ricompense in base a due passi successivi consente loro di risolvere problemi non banali, come nella situazione ambigua mostrata in fig. 13.

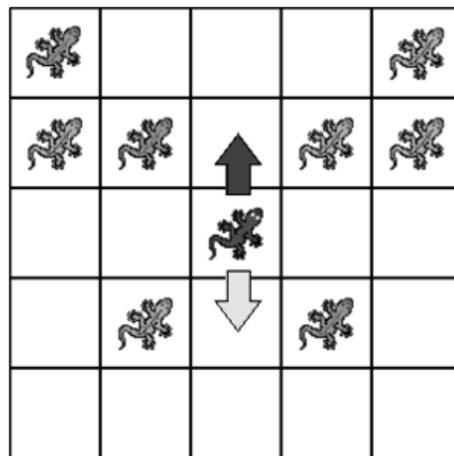


Fig. 13. Il camaleonte al centro della griglia (sarebbe rosso, gli altri sono tutti o verdi o blu) andando a sud compie una scelta preveggenente.

In questo caso il camaleonte potrebbe essere tratto in inganno dal fatto che la ricompensa al tempo  $t + 1$  è la stessa muovendo lungo le due direzioni indicate. Ipotizzando che questa situazione si riproponga un numero sufficiente di volte, l'algoritmo riuscirà a memorizzare il vantaggio del movimento verso sud, perché in quella direzione la ricompensa al tempo  $t+2$  sarà statisticamente maggiore. Ciò è dovuto alla forma, opportunamente scelta, della (2).

## BIBLIOGRAFIA

- ANDERSON P.W. (1972), *More is Different*, in «Science», 177(4047), pp.393-396.
- BELTRATTI, A., MARGARITA S., TERNA P. (1996), *Neural Networks for Economic and Financial Modelling*. ITCP, London.
- BLAKEMORE S.-J. E FRITH C. (2003), *Self-awareness and action*, in «Current Opinion in Neurobiology», 13, pp. 219-224.
- BONABEAU E., DORIGO M., THERAULAZ G. (1999), *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York.
- BOSTROM N. (2003), *Are You Living In a Computer Simulation?*, in «Philosophical Quarterly», 53(211), pp. 243-255. In linea a <http://www.simulation-argument.com/simulation.pdf>.
- CONTE R., TERNA P. (a cura di) (2000), *Una discussione sulla simulazione in campo sociale: mente e società*, in «Sistemi intelligenti», 12(2), pp. 326-331.
- DAWKINS R. (1999), *Foreword to Evolutionary Design by Computers*, in P.J. Bentley (a cura di), *Evolutionary Design by Computers*. Morgan Kaufmann, San Francisco.
- DONZELLI F. (1988) *Introduzione*, in F.A. von Hayek, *Conoscenza, mercato, pianificazione*. il Mulino, Bologna.
- FLAKE G.W. (1998), *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems and Adaptation*. MIT Press, Cambridge, MA. Codici di calcolo disponibili a <http://mitpress.mit.edu/books/FLAOH/cbnhtml/home.html>.
- GILBERT N., TERNA P. (2000), *How to build and use agent-based models in social science*, in «Mind & Society», n. 1, pp. 57-72.
- GILBERT N. E TROITZSCH K.G. (2005), *Simulation for the Social Scientist*. Open University Press, Buckingham.
- HAYEK F. (1952), *The Sensory Order*. University of Chicago Press, Chicago.
- HAYEK F.A. (1973), *Law, Legislation and Liberty*, volume I: Rules and Order. University of Chicago Press, Chigaco.
- HORGAN J. (1995), *From Complexity to Perplexity*, in «Scientific American», June.
- HORNIK K., STINCHCOMBE M., WHITE H. (1990), *Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks*, in «Neural Networks», pp. 551-560.
- LIN L.J. (1992), *Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching*, in «Machine Learning», 8, pp.293-321.
- KAELBLING L.P. E MOORE A.W. (1996), *Reinforcement Learning: A Survey* in «Journal of Artificial Intelligence Research», 4, pp. 237-285.
- KIRMAN A.P. (1992), *Whom or what does the representative individual represent?*, in «Journal of Economic Perspectives», 6, pp. 117-136.



MIGLINO O., DI FERDINANDO A, REGA A. E PONTICORVO M. (2007), *Le nuove macchine per apprendere: simulazioni al computer, robot e videogiochi multi-utente. Alcuni prototipi*, in «Sistemi intelligenti», 14(1), pp. 113-135.

PAGE S.E. (2007), *The difference: how the power of diversity creates better groups, firms, schools, and societies*, Princeton University Press, Princeton e Oxford.

SCHELLING T. (1978), *Micromotives and Macrobehavior*. Norton, New York.

SUTTON R.S. E BARTO A.G. (1998), *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

TERNA P. (1995), *Reti neurali artificiali e modelli con agenti adattivi*, in «Rivista italiana di economia», 0, pp. 71-106.

TERNA P. (2000), *Economic Experiments with Swarm: a Neural Network Approach to the Self-Development of Consistency in Agents' Behavior*, in F. Luna and B. Stefansson (a cura di), *Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming*. Kluwer Academic, Dordrecht and London, pp. 73-104.

TERNA P., BOERO R., MORINI M. E SONNESSA M. (a cura di) (2006), *Modelli per la complessità - La simulazione ad agenti in economia*. il Mulino, Bologna.

VRIEND N.J. (2002), *Was Hayek an Ace?*, in «Southern Economic Journal», 68(4), pp. 811-840.

WHITE H. (1989), *Learning in Artificial Neural Networks: A Statistical Perspective*, in «Neural Computation», 1, pp. 425-464.

ZIMMER C. (2006), *Alla ricerca del Sé*, in «Le scienze», 449, gennaio, pp.77-83. Traduzione da Zimmer C. (2005), *The Neurobiology of the Self*, in «Scientific American», November, pp. 92-101.

ZORZI M. (2006), *Dai neuroni al comportamento: la simulazione dei processi cognitivi con modelli generativi*, in «Sistemi Intelligenti», XVIII, 1, pp. 115-124.

---

Pietro Terna, Dipartimento di Scienze economiche e finanziarie G.Prato, corso Unione Sovietica 218bis, 10134 Torino, [terna@econ.unito.it](mailto:terna@econ.unito.it).

Riccardo Taormina, [ric.tao@gmail.com](mailto:ric.tao@gmail.com).

---

*Simulation models with intelligent agents: the surprising chameleons' world.*

*Abstract*

*From the beginning of the Nineties, an evident path of research in the field of artificial agents for social sciences, with a special reference to economics, has been that of employing neural nets as tools to build agents, initially also handling agents as standalone entities. In the middle of the Nineties, or a little after, models of agents able to act and interact were spreading in social sciences: that diffusion was particularly due to the rise of Swarm ([www.swarm.org](http://www.swarm.org)) and of the related protocol. In a parallel way, doubts about the plausibility of the neural models of agents were emerging out.*

*Now we can come back and work on a synthesis between neural nets and agent based model. We introduce here a few elements for a synthesis, with both an example based on the metaphor of interacting chameleons and the proposal of the Swarm protocol by means of Python ([www.python.org](http://www.python.org)), a powerful programming language relatively easy to learn, applying SLAPP (Swarm Like Agent Protocol in Python).*