



ἘΠΈΚΕΙΝΑ

International Journal of Ontology
History and Critics

RICCARDO TARANTINO

*Pattern emergenti nei sistemi biologici: la creatività intrinseca
dei modelli basati su agenti*

EPEKEINA, vol. 16, n. 1-2 (2023), pp. 1-14
Philosophy of Technology

ISSN: 2281-3209

DOI: 10.7408/epkn.

Published on-line by:

CRF – CENTRO INTERNAZIONALE PER LA RICERCA FILOSOFICA
PALERMO (ITALY)

www.ricercafilosofica.it/epekeina



This work is licensed under a Creative Commons
Attribution-NonCommercial-NoDerivs 3.0 Unported License.

Pattern emergenti nei sistemi biologici: la creatività intrinseca dei modelli basati su agenti

Riccardo Tarantino

Introduzione

Un algoritmo è, nell'opinione comune, quanto di più lontano si possa immaginare da qualunque tipo di *creatività*. La funzione degli algoritmi, infatti, è quella di produrre determinati *output* a partire da certi *input* in un numero finito di passi. Cosa può esserci di meno creativo di una sequenza di istruzioni non ambigue e dettagliate fino alla pedanteria impartite da un programmatore in un preciso linguaggio di programmazione?

Come intuibile, esistono in generale numerose definizioni di creatività, anche piuttosto diverse tra loro. In tempi recenti, la difficoltà di definire questo concetto è stata tra le motivazioni che hanno spinto alcuni ricercatori a cimentarsi finanche con una trattazione empirica del fenomeno, più specificamente di tipo neuroscientifico (Schubert 2021). Con la parola "creatività" si può intendere, per esempio, la capacità di pensare in maniera alternativa, l'abilità di scoprire nuovi problemi, il percorso che va da un'idea iniziale alla realizzazione di un prodotto finale e molto altro ancora. In generale, si tratta di una nozione indubbiamente correlata alla produzione di idee e artefatti in qualche modo utili e originali. Inoltre, viene spesso presentata come strettamente associata a una caratteristica prettamente umana come l'*intenzionalità* degli individui nell'atto di creare qualcosa di nuovo (Walia 2019).

Tuttavia, sembra che l'intenzionalità non sia una caratteristica *necessaria* del fenomeno creativo, come dimostrato (oltre che, in una certa misura, dall'esperienza quotidiana) anche da notevoli casi di creatività scientifica come quelli vissuti e raccontati dal grande fisico e matematico Henri Poincaré. Più in particolare, Poincaré descrisse diversi episodi creativi personali di carattere matematico riconducibili, a suo avviso, a dinamiche del tutto inconscie, seppur evidentemente indotte dalle conoscenze che aveva accumulato nel proprio campo di studi nel corso degli anni, in grado di conferire all'atto creativo le

caratteristiche di un processo in *background* (Rose 1999). Concentrarci su dei tipi di creatività non intenzionali (come quelli inconsci appena citati) ci fornisce uno spunto per interrogarci sulle forme assunte dalla creatività in un ambito non-umano. In particolar modo, ci permette di introdurre il dibattito sull'effettiva esistenza di un qualche tipo di creatività nelle macchine.

Questo articolo si occupa di indagare un tipo ancor più specifico di creatività, ossia la creatività inerente agli algoritmi utilizzati per l'implementazione dei *modelli basati su agenti*. Quello della modellizzazione basata su agenti è un particolare approccio scientifico basato sulla ricostruzione e sulla simulazione di fenomeni naturali attraverso l'interazione tra individui computazionali detti *agenti*, il cui comportamento viene codificato da un programmatore con l'utilizzo di algoritmi. Pur seguendo pedissequamente le istruzioni indicate dal programmatore, gli agenti si rivelano spesso in grado di generare schemi complessivi (o *pattern emergenti*) imprevedibili sulla base della sola analisi del comportamento del singolo e non deducibili dal codice a cui ciascuno di loro, inevitabilmente, si attiene.

Non a caso, questo tipo di modellizzazione viene usato per riprodurre al computer il funzionamento dei cosiddetti *sistemi complessi*, ossia sistemi privi di controllo centralizzato, in cui le parti sono fortemente interdipendenti e nei quali, pur mantenendosi una forte componente casuale, l'ordine e le regolarità emergono spontaneamente e si preservano grazie a un certo grado di resistenza alle perturbazioni e alla presenza di meccanismi di autoregolazione. La proprietà più determinante di questi sistemi, tuttavia, risiede nel fatto che i *pattern* emergenti che li caratterizzano sono determinati unicamente dall'articolata interazione tra le unità che li compongono, analogamente a quanto avviene nei modelli ad agenti (che sono così costruiti "ad immagine e somiglianza" dei sistemi reali che rappresentano).

Ora, senza volerci soffermare sulla questione di carattere più generale sul fatto che la natura possa essere "creativa" o meno, l'intento principale di questo lavoro è di mostrare come la *creatività computazionale* sia una caratteristica importante, se non addirittura intrinseca e necessaria, di questa categoria di modelli. Come vedremo, infatti, è *fondamentale* che un modello ad agenti ben implementato sia in grado di generare delle dinamiche globali emergenti che non siano in alcun modo inserite direttamente nel codice dal suo autore.

A dimostrazione di ciò, la prima sezione del contributo si sofferma sulla descrizione di un modello basato su agenti implementato con il *software* NetLogo (Wilensky 1999) e in grado di riprodurre alcuni degli aspetti salienti di uno specifico fenomeno biologico. In questa sezione, oltre a fornire una semplice descrizione di NetLogo, del modello e del processo simulato, approfondisco soprattutto il modo in cui gli agenti considerati riescono a produrre all'unisono dei *pattern* emergenti che non sono codificati dal programmatore, e rappresentano piuttosto il risultato spontaneo e imprevedibile a priori dell'interazione tra entità individuali dal comportamento algoritmico.

Nella seconda sezione, mi soffermo sugli aspetti teorici più generali e rilevanti presentati nella prima e cioè, sostanzialmente, sul perché l'approccio della modellizzazione ad agenti sia intrinsecamente così affine alla presenza di creatività computazionale, quasi che quest'ultima costituisca una vera e propria premessa e una garanzia metodologica di funzionamento del modello implementato.

Nella sezione conclusiva, sintetizzo il messaggio essenziale di questa incursione teorica nella modellizzazione basata su agenti e ne inquadro il valore nel contesto più generale della ricerca scientifica basata su simulazioni al computer.

1. Evoluzione *in silico*: un fenomeno emergente non codificato dal programmatore

L'*evoluzione biologica* è il cambiamento degli organismi al susseguirsi delle generazioni. Sebbene questo macrofenomeno si verifichi spesso in tempi estremamente lunghi (*troppo* lunghi per un osservatore umano), gli eventi che lo compongono sono, almeno in linea di principio, osservabili e possono in ogni caso essere inferiti o previsti (con un ampio margine di errore) attraverso l'uso di modelli teorici di tipo probabilistico. Questo cambiamento intergenerazionale, in estrema sintesi, non è infatti altro che l'effetto delle leggi dell'ereditarietà calate nel contesto delle interazioni ecologiche complesse tra gli organismi, senza dimenticare l'influenza notevole della componente casuale.

Sebbene esistano diversi motivi metodologici rilevanti per decidere di modellare un sistema biologico in evoluzione con delle simulazioni fatte al computer (Taylor & Jefferson 1995), quello più determinante resta, a mio avviso, quello appena menzionato riguardante l'enorme

lentezza di questo genere di processi nel “mondo reale”. Sebbene esistano anche importanti casi di studio dell’evoluzione di organismi reali condotti in laboratorio, si tratta normalmente di ricerche dalla durata davvero spropositata, svolgendosi nell’arco di decine di anni ed essendo per di più vincolate all’utilizzo pressoché esclusivo di specie dotate di brevissimi cicli vitali ed estremamente prolifiche (grazie, per esempio, a una modalità riproduttiva di tipo asessuale, come nel caso dei celebri batteri della specie *Escherichia coli*) (Lenski 2017).

Uno dei linguaggi di programmazione e ambienti di sviluppo integrato più suggestivi e di semplice utilizzo in grado di venire incontro ai ricercatori afferenti a quest’area della biologia è senz’altro NetLogo. NetLogo è specificamente pensato per l’implementazione e la simulazione di modelli basati su agenti, ed è oggi ampiamente utilizzato sia nell’ambito della didattica che della ricerca, grazie soprattutto all’intuitività della sua sintassi (pur essendo basato sui linguaggi Scala e Java, infatti, NetLogo è un linguaggio a sé stante) e alla possibilità di vedere facilmente in azione i sistemi computazionali costruiti passo dopo passo già in fase di implementazione attraverso la schermata di Interfaccia, con la quale l’utente interagisce per avviare e gestire le proprie simulazioni. Per usare dei termini un po’ meno specialistici, NetLogo è essenzialmente una piattaforma attraverso la quale è possibile programmare delle entità virtuali (gli agenti, appunto) e osservare come queste entità si comportino e interagiscano tra loro, in tempo reale, all’interno di un ambiente 2D.

Il *software* è inoltre associato a una *Models Library*, dalla quale è possibile selezionare e aprire un modello preesistente il cui funzionamento è verificato dai gestori di NetLogo. Una volta aperto il modello, è possibile interagire con esso tramite l’Interfaccia, comprenderne il funzionamento tramite una schermata Informazioni e, eventualmente, modificarlo per mezzo della schermata Codice. Come intuibile, è possibile imbattersi in una grande varietà di modelli riconducibili ad aree molto differenti, dalla filosofia alla chimica, dalla sociologia all’economia.

Ora, al fine di mostrare con un esempio concreto il tipo di creatività computazionale a cui faccio riferimento in questo articolo, intendo esporre brevemente un modello evolutivo tratto da questa libreria. Più nello specifico, il modello in questione riguarda la cosiddetta *microevoluzione*, che è l’oggetto di studio dell’area della *genetica di popolazioni*.

Questa disciplina si occupa di indagare le variazioni nel corso del tempo delle frequenze di determinate varianti di geni trasmesse, secondo le leggi dell'ereditarietà, da una generazione all'altra in un gruppo di organismi di una stessa specie e che condividono lo stesso areale geografico. Oltre a essere influenzate dai processi ereditari, queste frequenze sono soggette a oscillazioni indotte da un certo numero di meccanismi direzionanti, come per esempio la *selezione naturale*, che favorisce la replicazione delle varianti più "adatte" al contesto ecologico in cui gli organismi della popolazione si trovano.

Il nome del modello che ho scelto per supportare il mio argomento è altamente esplicativo: *Natural Selection – Camouflage* (Dabholkar & Wilensky 2020), ossia "Selezione Naturale – Mimetismo". Il modello rappresenta in maniera semplificata l'evoluzione di una popolazione di topi sottoposta a una pressione selettiva legata alla capacità di mimetizzarsi con l'ambiente. Prima di descrivere sinteticamente il funzionamento di questo sistema virtuale, è opportuno chiarire alcuni aspetti teorici di base.

Si noti preliminarmente come l'approccio computazionale consenta di trattare l'evoluzione di animali con regole di trasmissione dell'informazione genetica più complesse rispetto a quelle dei batteri, in quanto il genoma dei topi è organizzato in svariate coppie di cromosomi *quasi* identici tra loro detti *omologhi* (inoltre la specie non è asessuale ma presenta due sessi separati, l'uno in grado di produrre spermatozoi, l'altro cellule uovo). Questo significa che tutti i geni localizzati sui cromosomi sono presenti in doppia copia (escludendo i cromosomi sessuali, che qui non ci interessano). Ciò che gli autori del modello hanno fatto è stato considerare esclusivamente un gene ipotetico per il quale esistono soltanto due varianti alternative, *A* e *a*. Ciascun topo possiede un preciso *genotipo*, cioè una combinazione di queste due varianti geniche: in questo caso, i soli tre genotipi possibili sono *AA*, *Aa/aA* e *aa*. Il primo genotipo è detto *omozigote dominante*, il secondo *eterozigote* e il terzo *omozigote recessivo*. L'aspetto esteriore dei topi (il *fenotipo*) è qui determinato "rigidamente" dal genotipo. Più in particolare, il fenotipo "colore scuro del pelo" viene prodotto sia dal genotipo *AA* che dal genotipo *Aa* (o *aA*), mentre il fenotipo "colore chiaro del pelo" solamente dal genotipo *aa*.

Sofferamoci adesso sull'utilizzo del modello (vedi Figura 1 per visualizzarne l'Interfaccia): dopo aver impostato il numero iniziale di

topi per ciascun genotipo e per ciascun sesso, l'utente può decidere se attivare con un interruttore la predazione da parte di un volatile e regolare la probabilità di predazione (che determina il numero complessivo di predatori generati). È anche possibile generare in qualunque momento un nuovo individuo *mutante* con genotipo eterozigote premendo su un pulsante apposito. Dopo aver avviato la simulazione premendo un altro pulsante, si osservano i topi camminare casualmente nel mondo di simulazione di NetLogo, invecchiare e morire, o per via dell'età o a causa della predazione. Oltre che dal numero di predatori che si aggirano per il mondo di NetLogo, la probabilità di predazione per ciascun topo è influenzata dalla capacità di quest'ultimo di mimetizzarsi con l'ambiente circostante, cioè con il colore delle *patch* (le unità spaziali bidimensionali di NetLogo, ossia delle celle quadrate su un piano cartesiano) che lo circondano, che può essere più scuro o più chiaro e può essere regolato dall'utente anche nel corso della simulazione. In altri termini, i topi con colore del pelo più simile a quello dello sfondo sopravvivono (e possono riprodursi) più frequentemente quando incontrano lungo il loro cammino dei topi di sesso opposto, mentre quelli che risaltano di più sullo sfondo vengono uccisi più frequentemente, evento che riduce la rappresentanza nella popolazione delle varianti geniche di cui sono portatori.

Le capacità mimetiche sono dunque cruciali per canalizzare l'evoluzione del sistema in termini di frequenze di varianti geniche, dal momento che sono proprio i genotipi a determinare il colore del pelo e la selezione naturale agisce su questo carattere fenotipico. Ciascuna simulazione risulta enormemente velocizzata rispetto alla propria controparte reale: basti pensare al fatto che una singola unità di tempo discreta di NetLogo, il *tick*, viene considerata equivalente a una generazione di animali. L'andamento delle frequenze delle varianti geniche, dei genotipi e dei fenotipi è tracciato da dei monitor e da un grafico, dai quali è molto semplice constatare gli eventi di *fissazione* o *estinzione* di una delle due varianti nella popolazione. Questi eventi critici rappresentano il principale *fenomeno emergente* di questo sistema complesso biologico. L'interazione tra organismi, sia di tipo riproduttivo che predatorio, contribuisce insomma al cambiamento complessivo di una macro-entità integrata (la popolazione, appunto), fino al raggiungimento di stati irreversibili che contemplan la perdita di diversità genetica.

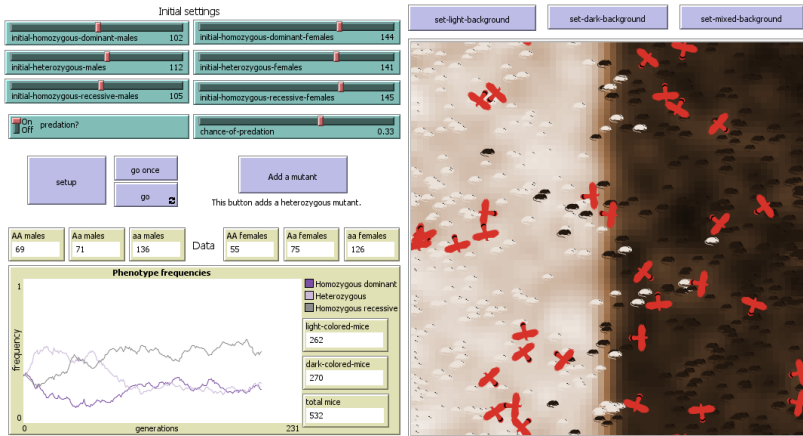


Figura 1: L'Interfaccia del modello implementato in NetLogo *Natural Selection - Camouflage* (Dabholkar & Wilensky 2020) nel corso di una simulazione. Si noti come sulla metà del mondo di simulazione più chiara siano presenti quasi esclusivamente topi di colore bianco e, viceversa, come la metà più scura sia popolata quasi interamente da topi neri, una conseguenza diretta del processo di selezione naturale basato sulle capacità mimetiche.

Ma il comportamento di questa macro-entità integrata è stato impostato dal programmatore? Esistono, in altre parole, delle righe di codice che *impongano* al sistema di evolvere in quel dato modo? La risposta è negativa, ed è facile dimostrarlo analizzando le varie porzioni di codice nel dettaglio. Nella schermata Codice corrispondente al modello appena descritto appaiono, in quest'ordine, le seguenti istruzioni: vengono definite le variabili "colore del pelo chiaro" e "colore del pelo scuro"; vengono definite le categorie di topi e predatori; vengono definite le caratteristiche dei topi (genotipo, età, partner, sesso); vengono definite le procedure di inizializzazione delle simulazioni, che generano gli agenti iniziali, sia topi che predatori, con le loro caratteristiche (genotipo, fenotipo, sesso, assenza di partner, età pari a 0, dimensioni, forma) e impongono il colore marrone allo sfondo; viene definita una procedura di simulazione composta da svariate sotto-procedure; vengono definite singolarmente le varie sotto-procedure, cioè quella che regola il movimento casuale degli agenti nel mondo, quella che regola la morte dei topi in base all'età, quella che regola la probabilità di morte dei topi in base alla presenza di predatori e alle proprie capacità mimetiche (in relazione al proprio fenotipo e al colore delle *patch* circostanti), quella che regola la ricerca di partner da parte dei topi e quella che regola la conseguente riproduzione in base alle regole probabilistiche dell'ereditarietà classica mendeliana; viene definita una procedura per l'aggiunta di individui mutanti eterozigoti alla pressione dell'apposito pulsante; viene definita una procedura per la gestione dinamica del numero di predatori; vengono definite le procedure per impostare il colore dello sfondo del mondo di NetLogo (chiaro, scuro o misto).

Come si può evincere dalla breve disamina appena compiuta, non vi è nel codice alcun riferimento, né esplicito né implicito, a istruzioni che il sistema nel suo complesso dovrebbe eseguire per generare il comportamento emergente delle fluttuazioni nel *pool* delle varianti geniche e della conseguente eliminazione della diversità genetica. L'attività di programmazione si sofferma *esclusivamente* sull'implementazione di semplici algoritmi assegnati alle varie classi di agenti, sia mobili (topi, predatori) che fissi (*patch*). Il processo di formazione delle dinamiche globali, esattamente come avverrebbe nel sistema biologico di riferimento, entra in funzione *spontaneamente* a partire dalle interazioni tra gli agenti. Eppure, tutto ciò che il modello è in grado di simulare discende, chiaramente, proprio da ciò che è scritto nella schermata

Codice. Questo apparente paradosso è, plausibilmente, l'aspetto in assoluto più interessante della modellizzazione basata su agenti, che prevede come suo stesso fondamento metodologico la produzione di sistemi computazionali in grado di determinare le proprie dinamiche in maniera autonoma, senza un intervento diretto del programmatore. A questo aspetto cruciale è dedicata la prossima sezione.

2. La creatività computazionale come criterio di validità dei modelli ad agenti

Una conclusione importante che discende direttamente dalla descrizione del modello appena commentato è che gli *output* dei modelli ad agenti possano rappresentare dei casi peculiari di creatività. La produzione spontanea di schemi emergenti (simulati) in qualche misura *nuovi, imprevedibili* e, spesso, *utili* per la ricerca scientifica, può infatti essere considerata una forma di creatività computazionale, in quanto risultante da una serie di algoritmi e, al contempo, non inferibile analizzando gli algoritmi implementati. Sulla base di questo presupposto, ciò che intendo sostenere in questa sezione è l'esistenza di un legame profondo tra creatività computazionale e buona pratica di costruzione dei modelli basati su agenti. Se, come sostengo, la soppressione di questa forma di creatività è addirittura *incompatibile* con la corretta progettazione di un modello, la creatività computazionale può essere ritenuta un requisito metodologico *essenziale* di questo particolare approccio di indagine. Ma procediamo con ordine.

L'aspetto probabilmente più pregnante dell'esempio precedente, che è in effetti generalizzabile a tutti i modelli basati su agenti in grado di rappresentare in maniera soddisfacente le dinamiche dei sistemi complessi a cui si ispirano, è la sua capacità di generare delle proprietà emergenti computazionali in maniera analoga a come queste si generano nella realtà. Vi è, in altri termini, un sostanziale *isomorfismo* tra la formazione delle proprietà emergenti generata dall'auto-organizzazione delle entità reali e la formazione delle stesse proprietà simulate nel modello computazionale, anch'esse generate dalla capacità degli agenti di auto-organizzarsi. In entrambi i casi, è del tutto assente un controllo centralizzato: le singole unità interagenti seguono delle leggi o si attengono a delle regolarità (che, nel caso del modello computazionale, corrispondono agli algoritmi sviluppati dal program-

matore), ma il risultato della loro attività collettiva viene generato solo a posteriori e senza alcun intervento diretto o legge di “livello superiore”.

Il passaggio dirimente di questo ragionamento sta nel fatto che, sebbene gli “ingranaggi” di questi sistemi siano ovviamente i singoli individui, l’evoluzione a livello sistemico aggiunga *qualcosa* rispetto ai vincoli di base posti dalle regole individuali, che sono puntiformi per definizione. Dal punto di vista dei singoli agenti, il codice scritto dal programmatore di ogni modello di questo tipo è chiaramente la sorgente di ciò che avviene nelle simulazioni. Tutto ciò che gli agenti sono in grado di fare è completamente basato sul programma, ed è proprio questo l’aspetto più sorprendente: se il modello è costruito correttamente (cioè, senza errori logici o di sintassi), gli individui computazionali eseguono automaticamente e alla perfezione tutti i comandi descritti nella schermata Codice, motivo per cui lo “scarto” in grado di generare i fenomeni emergenti non può che provenire dalle loro intricate interazioni. È proprio in questa fase che possono essere prodotti degli effetti imprevisti, in quanto il risultato di queste interazioni cumulative non è e *non dovrebbe* essere descritto in alcun modo dall’autore del modello.

Quest’ultimo approccio, plausibilmente, reprimerebbe infatti il fattore imprevedibilità in favore di un maggiore (o totale) controllo sul sistema, un’operazione in piena contraddizione con le stesse fondamentali teoriche della modellizzazione basata su agenti. In linea di principio, si potrebbe senz’altro decidere di implementare un modello in cui anche il comportamento aggregato del sistema sia puntigliosamente definito nel codice. Inserire delle regole di livello superiore, tuttavia, infrangerebbe l’isomorfismo citato precedentemente, in quanto non sarebbero più, o non sarebbero soltanto gli agenti ad auto-organizzarsi e a produrre i *pattern* emergenti, ma sarebbe piuttosto uno specifico centro di controllo a “imporre” tutto dall’alto (il programmatore). In altre parole, incrementare il determinismo intervenendo direttamente al livello *target* piuttosto che a quello inferiore ridurrebbe drasticamente non solo la componente stocastica (che è invece di primaria importanza nei sistemi complessi), ma anche e soprattutto l’aspetto dell’auto-organizzazione. In sintesi, un modello così concepito sarebbe in contraddizione con i principi della modellizzazione basata su agenti e quindi, tecnicamente, *non sarebbe* più un modello basato su agenti. Cosa altrettanto importante, non conterrebbe alcun elemento creativo:

la novità normalmente generata dalla simulazione verrebbe rimpiazzata da un risultato noto (anche solo in maniera approssimativa) fin dall'inizio.

Questa è una delle ragioni per cui impedire a un modello ad agenti di “essere creativo” sarebbe altamente invalidante da una prospettiva prettamente epistemologica. Consideriamo nuovamente il modello ipotetico descritto poc'anzi, in cui è il programmatore a codificare direttamente anche le regole di livello superiore del sistema. Un modello limitato sul versante del proprio contributo creativo endogeno sarebbe inevitabilmente il risultato del contributo (sostitutivo) del programmatore nella produzione degli *output* delle simulazioni. Naturale conseguenza di questa modellizzazione “intrusiva” sarebbe la produzione di risultati artefatti e, molto verosimilmente, distorti (nell'accezione del termine inglese *biased*) dalle aspettative e dalle previsioni del ricercatore. Non è un caso che la metodologia classica della modellizzazione ad agenti preveda, in seguito alla corretta implementazione del modello, una fase più o meno stringente di *validazione* (Wilensky & Rand 2015, 325-326): quando si possiede un modello completo e funzionante, infatti, è auspicabile svolgere delle comparazioni, qualitative o quantitative, atte a *dimostrare* l'aderenza degli *output* delle simulazioni svolte con le osservazioni, gli esperimenti o le deduzioni matematiche già disponibili per quel fenomeno. È grazie al processo di validazione che è possibile accertare che gli *output* del proprio modello siano consistenti con quelli del sistema che esso mira a rappresentare, ed è solo *dopo* questa fase che è possibile utilizzare il modello per compiere delle investigazioni su aspetti del fenomeno considerato non ancora chiari o difficili da trattare con altri metodi o approcci. Codificare il comportamento emergente del sistema nello stesso modo in cui vengono codificati i comportamenti dei singoli componenti invertirebbe completamente l'ordine del processo, rendendo inaffidabili sia i risultati che il modello che li produce.

Esiste, inoltre, almeno un'altra motivazione epistemologica determinante a favore di quello che potremmo definire il “criterio di creatività” dei modelli ad agenti, e risiede nella correlazione positiva tra creatività e *semplicità*. Nel modello ipotetico in cui l'*output* non è emergente ma postulato dal programmatore, è chiaramente necessario inserire dei passaggi ulteriori rispetto a quelli che basterebbero se si decidesse di codificare esclusivamente il comportamento degli agenti,

il che implica un minor grado di semplicità. Un livello di sofisticazione più elevato è normalmente deleterio nella ricerca scientifica, perché contrario al *principio di parsimonia* già noto fin dai tempi di Guglielmo d'Occam: l'approccio di modellizzazione più funzionale e interessante è quello che si avvale del minor intervento possibile da parte dello scienziato. Nel caso dei modelli basati su agenti, la quantità ideale di interventi da parte del programmatore dovrebbe essere molto ridotta, e tali interventi dovrebbero essere limitati agli aspetti più facilmente determinabili e descrivibili del sistema, ossia al comportamento dei singoli agenti (se sono disponibili sufficienti dati empirici al riguardo).

Alla luce di quanto appena argomentato, lungi dal risultare un errore o un'imprecisione in fase di codifica, la riproduzione *in silico* dell'auto-organizzazione tipica dei sistemi complessi reali è, a mio avviso, una possibile forma di creatività computazionale importantissima e in grado di ricoprire il ruolo di criterio di demarcazione metodologico. La possibilità di riprodurre schemi emergenti in un sistema digitale in maniera indiretta, cioè usando degli algoritmi assegnati ai soli agenti, è ciò che più avvicina funzionalmente le simulazioni fatte al computer ai processi naturali modellati, preservando al massimo grado l'isomorfismo tra modello e realtà.

In sintesi, una volta trasposta correttamente in un sistema computazionale, la creatività spontanea presente nei sistemi complessi reali può essere considerata sintomatica di una corretta impostazione del lavoro di implementazione, in quanto riduce il contributo del ricercatore al minimo indispensabile e protegge da distorsioni soggettive e direzionamenti esterni.

Conclusioni

Le simulazioni prodotte dai modelli ad agenti, pur nella loro totale mancanza di intenzionalità, sono in grado di produrre degli *output* che sarebbe arduo non definire creativi. Una mirabile conclusione di questa indagine è l'osservazione di come l'intelligenza artificiale sia in grado di replicare dei frammenti di natura in modo analogo a come sembra farlo la natura stessa, e cioè, lasciando che i propri componenti si organizzino autonomamente in innumerevoli configurazioni inaspettate. Il tassello che ho tentato di aggiungere con questo contributo è la constatazione del fatto che, oltre a possedere un indiscutibile fascino

intrinseco, questo profondo isomorfismo possa inoltre essere considerato un criterio di demarcazione cruciale per distinguere un buon modello basato su agenti da uno soggetto a limitazioni e *bias* importanti. Questa riflessione, di per sé piuttosto specifica e, per certi versi, anche abbastanza tecnica, ci spinge a ragionare su aspetti ben più generali del concetto di creatività. Sia nel mondo reale che in quello digitale (sempre che una distinzione del genere abbia davvero un senso), la creatività sembra configurarsi come un “effetto collaterale” del raggiungimento di una soglia critica di complessità. Il fatto che questa soglia possa essere raggiunta seguendo due vie che sembrano essere totalmente alternative tra loro, come le leggi e le regolarità della natura, da una parte, e la metodologia scientifica e l’approccio di programmazione, dall’altra, merita a mio avviso di essere indagato molto più ampiamente e costituisce, se non una prova, perlomeno un indizio del fatto che delle forme di creatività possano emergere indipendentemente dal sostrato considerato.

Riferimenti bibliografici

Dabholkar, S. & Wilensky, U. 2020, *NetLogo Natural Selection - Camouflage model*. Url: <http://ccl.northwestern.edu/netlogo/models/NaturalSelection-Camouflage>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Lenski, R. 2017, *Experimental evolution and the dynamics of adaptation and genome evolution in microbial populations*, “International Society for Microbial Ecology”, 11, pp. 2181-2194.

Rose, D. 1999, *Creativity, intentionality and the conscious/unconscious distinction: a neural theory*, “Journal of Intelligent Systems”, 9(5-6), pp. 407-442.

Schubert, E. 2021, *Creativity is optimal novelty and maximal positive affect: a new definition based on the spreading activation model*, “Frontiers in Neuroscience”, 15, 612379.

Taylor, C. & Jefferson, D. 1995, *Artificial Life as a tool for biological inquiry*, in *Artificial Life: an overview*, a cura di Christopher Langton,

MIT Press, Cambridge (Massachusetts), pp. 1-13.

Walia, C. 2019, *A dynamic definition of creativity*, “Creativity Research Journal”, 31(3), pp. 237-247.

Wilensky, U. 1999, NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U. & Rand, W. 2015, *An introduction to agent-based modeling*, MIT Press, Cambridge (Massachusetts).

Riccardo Tarantino
Università degli Studi di Palermo
riccardo.tarantino01@unipa.it