Applicazione di algoritmi genetici a contesti videoludici

Lorenz Cuno Klopfenstein Piervincenzo Russo Alessandro Manini
Alessandro Bogliolo Emanuele Lattanzi

STI — Istituto di Scienze e Tecnologie
Università degli Studi di Urbino, 61029 Urbino, Italia

Sommario Benché molti videogiochi esibiscano comportamenti apparentemente intelligenti, creando schemi di gioco sempre nuovi e imprevedibili e opponendo al giocatore antagonisti capaci di prendere decisioni autonome e adeguate ad ogni contesto di gioco, raramente tali comportamenti sono determinati da vere e proprie applicazioni di intelligenza artificiale.
Questo lavoro prende in esame le possibili applicazioni di algoritmi genetici in contesti videoludici, dandone una classificazione di massima e presentando esempi concreti sviluppati a scopo sperimentale.

1 Introduzione

Il mercato videoludico ha registrato nel 2006 un fatturato di circa 7 miliardi di Dollari nei soli Stati Uniti [1] e si prospetta che in un prossimo futuro possa arrivare a competere con l'industria cinematografica [2,4]. I videogiochi coinvolgono un numero molto elevato di giocatori, rappresentano un'industria in costante espansione e si prestano a sperimentare l'applicazione di teorie e tecniche innovative sviluppate in ambito accademico [5]. Particolarmente evidenti sono i continui sviluppi sul fronte della grafica tridimensionale in tempo reale, che ha raggiunto livelli di fotorealismo molto convincenti anche su *hardware* di consumo. Per contro, sono ancora relativamente poco sfruttate nell'ambito dei videogiochi le potenzialità delle tecniche di *machine learning*, che potrebbero permettere ad un videogioco di adattarsi autonomamente alle situazioni di gioco e, soprattutto, di apprendere dal giocatore stesso nuovi comportamenti e strategie.

Nella maggior parte dei videogiochi allo stato dell'arte i meccanismi di gioco sono preprogrammati direttamente dagli sviluppatori o predefiniti mediante l'uso di varie tecniche, tra cui anche le reti neurali (un approccio utilizzato recentemente è quello di far sì che il prototipo di avversario impari ad emulare le azioni di un giocatore esperto durante la fase di sviluppo, per poi replicare questi comportamenti nel gioco finale). Nelle fasi di gioco vere e proprie gli avversari simulati si comportano secondo una logica statica di azione-reazione: il videogioco oppone una reazione predeterminata alle azioni del giocatore, comportandosi più come un sistema esperto che come un'entità di intelligenza artificiale, il cui comportamento complesso dovrebbe invece scaturire autonomamente dall'applicazione di regole di apprendimento.

Dal punto di vista del gioco, l'assenza di capacità di apprendimento e di meccanismi evolutivi fa sì che quando il giocatore individua una debolezza in uno schema di gioco, possa sfruttarla ripetutamente senza che lo schema cambi per porvi rimedio [6]. In particolar modo questo vale per giochi in tempo reale che richiedono decisioni veloci

in un ambiente dinamico ed ostile, mentre nel caso di giochi a turni e senza limiti di tempo stringenti (per esempio gli scacchi o giochi strategici a turni) esistono sistemi di IA capaci di competere con giocatori umani anche per mera enumerazione delle azioni possibili [3].

2 Algoritmi genetici nei videogiochi

Gli algoritmi genetici sono basati sulla metafora dell'evoluzionismo, che non tiene in considerazione la capacità di apprendimento dall'esperienza di ogni singolo individuo di una popolazione, quanto piuttosto la tendenza dell'intera popolazione a produrre individui sempre più adatti a sopravvivere nel contesto in cui sono inseriti. Tale capacità di adattamento, se attribuita all'intera popolazione, è la manifestazione dell'intelligenza artificiale che l'algoritmo genetico induce su base statistica.

L'applicazione di algoritmi genetici ai videogiochi richiede l'individuazione di gradi di libertà che consentano l'evoluzione degli schemi di gioco, la definizione di un genoma in grado di codificare tali gradi di libertà, la definizione di una funzione di *fitness* valutabile durante il gioco e l'individuazione di paradigmi di gioco in grado di produrre dati statisticamente significativi e utili all'evoluzione in tempi compatibili con quelli del gioco.

Sulla scorta di queste osservazioni, è possibile classificare i meccanismi di gioco basati su algoritmi genetici in due macro-categorie: meccanismi *evolutivi* che espongono il motore genetico permettendo al giocatore di confrontarsi con l'algoritmo genetico nella ricerca del genotipo più adatto al contesto di gioco; meccanismi *adattativi* che utilizzano il motore genetico per adattare gli schemi di gioco alle strategie e all'abilità dal giocatore.

3 Giochi evolutivi

La base per lo sviluppo di giochi evolutivi è un motore genetico con un'interfaccia utente che permetta al giocatore di seguire l'evoluzione generazione dopo generazione, di osservare il comportamento e la *fitness* dei singoli individui e di compiere azioni che lo rendano in grado di condizionare l'evoluzione stessa. Lo scopo del gioco è la creazione o l'individuazione di elementi della popolazione iniziale la cui discendenza sia massimamente rappresentata nella popolazione finale. Le diverse tipologie di gioco che rientrano in questa categoria sono contraddistinte dal tipo di azioni che il giocatore può intraprendere per specificare o selezionare gli individui su cui puntare.

– **Sintesi del genotipo.**
  Il giocatore specifica il genotipo di un individuo selezionando ad uno ad uno gli alleli di ogni gene. L'individuo sintetico è inserito nella popolazione iniziale generata casualmente e sottoposto insieme ad essa al motore evolutivo. L'algoritmo genetico tiene traccia della discendenza dell'individuo sintetico in modo da determinare, ad ogni generazione, due parametri che determinano lo *score* del giocatore: la percentuale di popolazione che discende dall'individuo sintetico e il grado di somiglianza tra il genotipo sintetico iniziale e quello dell'individuo migliore della generazione corrente.

– **Scommessa.**
  Il giocatore dispone di una quantità di denaro iniziale che può utilizzare per puntare su uno o più individui della popolazione iniziale di cui conosce il livello di *fitness*. Come in qualsiasi sistema di scommesse, puntare sugli individui favoriti (con *fitness* maggiore) comporta una minor ricompensa in caso di vincita. L'algoritmo genetico tiene traccia, generazione dopo generazione, della discendenza di ciascun individuo su cui il giocatore ha puntato. Il punteggio finale è determinato moltiplicando ogni puntata per il numero di discendenti dell'individuo nella popolazione finale e per una quotazione inversamente proporzionale alla *fitness* dell'individuo su cui si era puntato.
– **Selezione della casta.**
  In questo caso il giocatore dispone di un numero limitato di punti che può spendere per esaminare la *fitness* degli individui di prima generazione o per selezionare quelli da includere nella propria "casta". L'algoritmo genetico evolve mantenendo separati gli individui della casta dai restanti individui, ma sottoponendoli allo stesso meccanismo di selezione naturale. In pratica, il crossover avviene separatamente all'interno e all'esterno della casta, ma la selezione avviene sull'intera popolazione. Il punteggio è proporzionale alla dimensione della casta nella popolazione finale.

4 Giochi adattativi

Nei giochi adattativi il motore genetico viene utilizzato in modo trasparente all'utente per far evolvere sulla base dell'esperienza lo schema di gioco o l'abilità degli antagonisti che il gioco oppone al giocatore. In ogni caso l'obiettivo dell'algoritmo genetico è quello di massimizzare il successo delle azioni antagoniste opposte al giocatore. Nell'ambito dell'algoritmo genetico, il giocatore rappresenta quindi il contesto al quale l'individuo, o la popolazione di individui, devono adattarsi. La velocità di questo processo evolutivo dipende dal tempo richiesto per valutare la *fitness* degli individui antagonisti.

Nel corso di una partita il giocatore può fronteggiare un solo individuo o un'intera popolazione di individui. Nel primo caso, ogni partita consentirà di valutare la *fitness* di un singolo individuo, mentre nel secondo potrà consentire la valutazione della *fitness* di un'intera popolazione di individui, rendendo più veloce l'evoluzione degli antagonisti.

– **Giocatore vs. popolazione:** lo schema di gioco mette il giocatore a confronto con tutti gli elementi della popolazione antagonista, dei quali viene valutata la *fitness* al termine della partita in base al successo dell'azione antagonista che sono riusciti ad esercitare. La popolazione evolve al termine dello schema presentando al giocatore una nuova generazione di antagonisti.
– **Giocatore vs. individuo:** lo schema di gioco oppone al giocatore un solo individuo alla volta la cui *fitness* viene valutata al termine della partita. Ogni passo evolutivo richiede il completamento di tante partite quanti sono gli individui della popolazione.
– **Giocatori multipli vs. individuo:** lo schema di gioco oppone più giocatori (contemporaneamente o separatamente) ad ogni individuo della popolazione. La *fitness*

dell'individuo viene valutata su base statistica come media del successo ottenuto confrontandosi con diversi giocatori. L'evoluzione avviene quando è stata valutata la *fitness* di tutti gli individui della popolazione.

Indicando con N la dimensione della popolazione, con n il numero di individui che intervengono in uno stesso schema di gioco, con p il numero di partite necessarie a valutare la *fitness* di ogni individuo, il numero di partite necessarie a raccogliere dati sufficienti a produrre una nuova generzione di antagonisti è dato da $N/n * p$. Nel primo caso $n = N$ e $p = 1$, per cui basta una sola partita per produrre una nuova generazione. Nel secondo caso $n = 1$ e $p = 1$, per cui servono N partite per produrre una nuova generazione. Nel terzo caso $n = 1$ e $p > 1$ per cui servono Np partite per generazione.

Nel caso in cui l'evoluzione richieda più di una partita per generazione, è opportuno che le informazioni necessarie alla valutazione della *fitness* vengano raccolte da partite condotte indipendentemente da più giocatori, nell'intento di riportare il tempo di evoluzione percepito da ogni giocatore entro la durata di una partita.

È utile osservare che la velocità con cui si susseguono le generazioni non necessariamente corrisponde alla velocità di convergenza dell'algoritmo genetico verso una popolazione di antagonisti imbattibili. Infatti, l'efficacia dell'evoluzione dipende dalla natura del problema e del comportamento del giocatore (o dei giocatori) da fronteggiare. Se l'algoritmo genetico viene esposto ad un giocatore che adotta sempre la stessa strategia di gioco la popolazione di antagonisti non ha modo di sviluppare difese efficaci ad altre strategie. Questa osservazione è alla base di meccanismi di gioco ad evoluzione rapida che permettono al giocatore di fingere una strategia per sbilanciare la reazione degli antagonisti e infine adottare una strategia opposta per batterli.

5 Prototipi

Tutti i paradigmi di gioco discussi nelle sezioni precedenti sono stati implementati e testati.

5.1 Prototipi di giochi evolutivi

Per l'implementazione dei giochi evolutivi è stato sviluppato un package Java che offre classi, interfacce e metodi che implementano il motore genetico e i meccanismi di base per la gestione della sintesi del genotipo, delle scommesse e dei meccanismi di selezione. Il package consente l'applicazione di tali meccanismi a qualsiasi problema, o a qualsiasi gioco, che può essere implementato specificando il genoma, la regola di decodifica e la funzione di *fitness*.

A scopo esemplificativo il package è stato utilizzato per lo sviluppo di un semplice gioco evolutivo il cui protagonista è un topolino che si muove in uno spazio bidimensionale quadrato popolato da un cane (che si muove in modo casuale) e da un gatto (che si muove nel tentativo di mangiare il topo e di tenersi alla larga dal cane). Il topolino si muove in base ad un istinto dettato dal suo genotipo. Ogni gene del genoma codifica la reazione del topolino ad una particolare situazione, rappresentata dalle posizioni relative di cane e gatto. Ad esempio, il gene che codifica la reazione del topolino alla

situazione in cui il cane sia alla sua sinistra e il gatto alla sua destra ha tre soli alleli: *left*, che lo induce ad andare a sinistra, *halt*, che lo induce a non muoversi, e *right*, che lo induce ad andare a destra. Il genoma è composto da 18 geni, con 3 alleli ciasuno. La *fitness* di ogni individuo della popolazione di topolini è valutata simulando l'azione e calcolando il numero di mosse del topolino prima di essere mangiato dal gatto. L'evoluzione insegna al topolino, generazione dopo generazione, a difendersi dal gatto seguendo le mosse del cane.

5.2 Prototipi di giochi adattativi

Il primo prototipo di gioco adattativo è stato sviluppato a partire dal gioco evolutivo del topolino, affidando il controllo del gatto ad un giocatore. Poichè nel corso di una partita il giocatore incontra un solo topolino la cui *fitness* viene valutata nell'ambito di una sola partita, il gioco rientra nella seconda tipologia di giochi adattativi, secondo la classificazione proposta nella sezione precedente.

Inoltre sono state sviluppate due applicazioni stand-alone, *Genetic RTS* e *Genetic Platform*, come esempi di giochi adattativi in cui il giocatore fronteggia un'intera popolazione di antagonisti, facendo evolvere il gioco ad ogni schema.

Genetic RTS è un gioco di strategia in tempo reale, in cui il giocatore ha il compito di difendere una base situata al centro di una mappa bidimensionale disponendo attorno ad essa un numero limitato di strutture difensive. Ogni struttura difensiva ha caratteristiche peculiari che la rendono più o meno efficace nel difendere l'avanzata di determinate tipologie di nemici. I nemici sono gli individui della popolazione genetica, caratterizzati da un genoma che ne determina il tipo (aerei, carri armati...), la velocità di movimento, la resistenza e la direzione di attacco.

(a) Screenshot di "Genetic RTS". In evidenza le difese della base (piazzate dal giocatore umano) e le unità attaccanti (determinate dalla popolazione genetica).

(b) Dopo alcune generazioni, la strategia di attacco si conforma alla base del giocatore: la maggior parte delle unità attacca da nord-est, dove sono assenti difese contro i carri armati.

Il gioco è strutturato in un'alternanza di fasi di costruzione (o modifica) delle strutture difensive da parte del giocatore e di attacco da parte di una popolazione di nemici (che entrano nello schema in rapida successione). La *fitness* delle unità nemiche viene calcolata al termine della fase di attacco valutando quanto sono riuscite ad avvicinarsi alla base prima di essere abbattute e premiando il fatto che siano riuscite a colpire la base. Ogni fase di gioco manda in campo una nuova generazione di nemici. L'algoritmo genetico adatta progressivamente la popolazione di nemici alle difese installate dal giocatore, che può eventualmente modificarle per disorientare gli avversari o per rimediare a delle vulnerabilità.

Ogni posizionamento e riposizionamento delle strutture difensive ha un costo per il giocatore, che inizia il gioco con un budget limitato e continua a giocare fintanto che la base resiste agli attacchi.

Genetic Platform è un classivo videogioco di tipologia *platform* in cui il protagonista si muove in un mondo bidimensionale popolato da nemici di vario genere. Ogni nemico è un individuo di una popolazione genetica il cui genoma ne codifica:

- la tendenza a stazionare in prossimità dei *bonus* della mappa;
- la tendenza a seguire le mosse del protagonista;
- la capacità, l'altezza e la frequenza di salto;
- la velocità di spostamento orizzontale;
- l'altezza di volo.

La *fitness* di ogni individuo viene calcolata in tempo reale (e può essere visualizzata in sovraimpressione durante il gioco) sulla base del tempo di permanenza dell'individuo in prossimità del giocatore. Questo fattore infatti è un buon indicatore dell'efficacia dell'azione antagonista svolta dell'individuo. La *fitness* viene inoltre incrementata se l'individuo riesce ad uccidere il giocatore e decrementata se viene a sua volta ucciso.

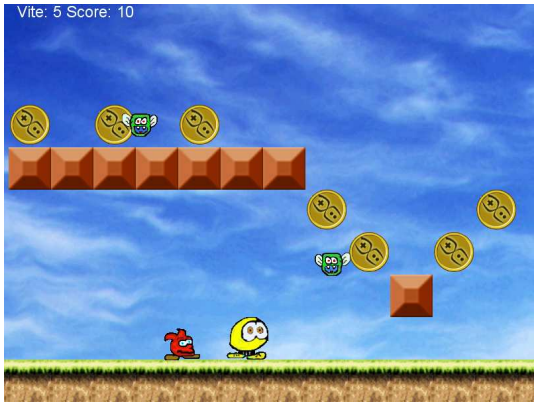

Figura 1. Screenshot di "Genetic Platform".

5.3 Considerazioni finali

I prototipi implementati hanno permesso di verificare l'applicabilità degli algoritmi genetici nel contesto nei videogiochi al duplice scopo di permettere un'interazione diretta dell'utente con il motore evolutivo (nel caso di giochi evolutivi) e di adattare dinamicamente gli schemi di gioco alla crescente abilità dei giocatori (nel caso di giochi adattativi). In quest'ultimo caso, è stato verificato il corretto funzionamento dell'algoritmo genetico e quindi l'adattamento delle strategie di gioco a quelle del giocatore. In particolare per quanto riguarda "Genetic RTS" è stato possibile apprezzare la progressiva focalizzazione degli attacchi nemici in zone vulnerabili della base del giocatore, anche nel giro di poche generazioni di individui (mediamente 3).

L'attività di ricerca attualmente in corso è finalizzata a valutare precisamente l'efficacia dei meccanismi di gioco in termini di usabilità e di capacità di adattamento all'utente. A tal fine tutti i prototipi sviluppati saranno distribuiti ad un campione di utenti controllato e resi disponibili in rete.

Riferimenti bibliografici

1. "Essential Facts about the computer and video game industry". Entertainment Software Association. 2006. `http://www.theesa.com`
2. E. Nussenbaum. "Video Game Makers Go Hollywood. Uh-Oh.". New York Times, 22 Agosto 2004.
3. M. Buro, Timothy M. Furtak. "RTS Games and Real-Time AI Research". IJCAI-2003 poster session.
4. P. Thurrott. "Top stories of 2001, #9: Expanding video-Game market brings microsoft home for the holidays". Windows & .NET Magazine Network. 2002.
5. Aliza Gold. "Academic AI and Video games: a case study of incorporating innovative academic research into a video game prototype". Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05). 2005.
   `http://nn.cs.utexas.edu/downloads/papers/gold.cig05.pdf`
6. Mark Wars. "Computer games get smarter". BBC News Online technology.
   `http://news.bbc.co.uk/2/hi/entertainment/1237848.stm`