

**UNIVERSITÀ CARLO CATTANEO - LIUC**

Scuola di Ingegneria industriale

Corso di laurea in Ingegneria gestionale - Classe L-9

**Studio e Applicazione di Reinforcement  
Learning in Modelli Epidemiologici  
Multi-Agente**

Relatore: Prof. Francesco Bertolotti

Tesi di laurea di:

Niccolò Kadera

Matricola n° 0028544

Anno Accademico 2023 - 2024

## **Autorizzazione alla consultazione della tesi di laurea e trattamento dei dati personali**

Il sottoscritto Niccolò Kadera matricola n° 0028544 nato a Legnano (MI) il 26 settembre 2002 autore della tesi di Laurea dal titolo Studio e Applicazione di Reinforcement Learning in Modelli Epidemiologici Multi-Agente:

- Autorizza la consultazione della tesi stessa, fatto divieto di riprodurre, in tutto o in parte, quanto in essa contenuto.
- Autorizza per quanto necessita l'Università Carlo Cattaneo - LIUC ai sensi della legge n. 196/2003 al trattamento, comunicazione diffusione e pubblicazione in Italia e all'estero dei propri dati personali per le finalità ed entro i limiti illustrati dalla legge.

Dichiara, inoltre:

- Di aver preso atto che, ai sensi della legge 19 aprile 1925, n. 475 tutt'ora in vigore, per gli elaborati presentati agli esami per il conseguimento della laurea si configura il reato di plagio e che, qualora gli elaborati fossero opera, anche parziale di altri e quindi si manifestasse il plagio, il reato è punito con la reclusione da tre mesi ad un anno.
- Di aver consegnato una copia dell'elaborato al Relatore e all'eventuale Correlatore;
- Di aver caricato l'elaborato e il riassunto sul proprio Self Service Studenti;
- Che l'elaborato in formato elettronico è conforme alla copia a stampa in tutte le sue parti;
- Di non avere libri in prestito dalla biblioteca Mario Rostoni;
- Di essere a conoscenza della normativa antiplagio e di essere consapevole che l'elaborato sarà sottoposto al vaglio antiplagio del software "Turnitin".

Consapevole che chiunque rilasci dichiarazioni mendaci è punibile ai sensi del codice penale e delle leggi speciali in materia, ai sensi e per gli effetti degli art. 75 e 76 dpr 445/2000.

Firma: Niccolò Kadera

# Indice

<b>1 Abstract</b>	<b>1</b>
<b>Capitolo: 1 Introduzione all'Agent</b>	
<b>Based Modelling e al Reinforcement Learning</b>	<b>2</b>
<b>2 ABM</b>	<b>2</b>
2.1 Introduzione . . . . .	2
2.1.1 Struttura . . . . .	2
2.1.2 Principi operativi dell'Agent-Based Modelling . . . . .	3
2.2 Agenti . . . . .	3
2.2.1 Fenomeno emergente . . . . .	4
2.2.2 Caratteristiche degli agenti . . . . .	5
2.3 Vantaggi . . . . .	6
2.4 Svantaggi . . . . .	7
2.5 Metodi per la descrizione . . . . .	7
2.5.1 Introduzione ODD . . . . .	8
2.5.2 Composizione ODD . . . . .	8
<b>3 Reinforcement Learning</b>	<b>11</b>
3.1 Introduzione . . . . .	11
3.2 Modello . . . . .	11
3.3 Q-Table . . . . .	12
3.4 Ricompensa e aggiornamento q-table . . . . .	13
3.5 Esplorazione o Sfruttamento (Exploration or Exploitation) . . . . .	14
3.5.1 Esplorazione . . . . .	14
3.5.2 Sfruttamento . . . . .	15
3.5.3 Trade-off . . . . .	15
3.6 Algoritmo di Epsilon-Greedy . . . . .	15
<b>Capitolo: 2 Implementazione di ABM</b>	
<b>e Reinforcement Learning</b>	<b>18</b>

<b>4 ABM Epidemiologico</b>	<b>18</b>
4.1 Introduzione . . . . .	18
4.2 Panoramica . . . . .	18
4.2.1 Scopo . . . . .	18
4.2.1.1 El Farol Bar . . . . .	18
4.2.1.2 Analisi di sensibilità El Farol . . . . .	20
4.2.1.3 Estensione epidemiologica . . . . .	21
4.2.1.4 Analisi di sensibilità epidemia . . . . .	23
4.2.1.5 Agenti . . . . .	25
4.2.1.6 Policy Maker . . . . .	27
4.2.2 Variabili di stato . . . . .	30
4.2.2.1 Variabili dell'agente . . . . .	30
4.2.3 Variabili globali . . . . .	31
4.2.3.1 Variabili del modello . . . . .	31
4.2.3.2 Parametri modello . . . . .	31
4.2.4 Descrizione dei processi e composizione . . . . .	32
4.2.4.1 Diffusione dell'epidemia . . . . .	32
4.2.4.2 Strategia dell'agente . . . . .	34
4.2.4.3 Presenza al bar . . . . .	35
4.2.4.4 Gestione delle strategie del Policy Maker . . . . .	37
4.2.4.5 Calcolo del costo degli infetti e azioni del Policy Maker . . . . .	38
4.2.4.6 Ciclo del modello e flusso generale . . . . .	39
4.3 Concetti di design . . . . .	40
4.3.1 Decorso del livello di contagio . . . . .	40
4.3.2 Stocasticità . . . . .	41
4.3.3 Ulteriori fenomeni emergenti . . . . .	41
4.4 Dettagli . . . . .	42
4.4.1 Inizializzazione . . . . .	42
4.5 Risultati . . . . .	43
4.5.1 Introduzione . . . . .	43
4.5.2 Fenomeni emergenti . . . . .	44
4.5.2.1 Effetti del contagio sulla presenza al bar . . . . .	44
4.5.2.2 Impatto del policy maker sulla presenza al bar . . . . .	46

4.5.3 Confronto con altri modelli . . . . .	47
<b>5 Reinforcement Learning in ABM</b>	<b>50</b>
5.1 Introduzione . . . . .	50
5.2 Panoramica Q-Table . . . . .	50
5.2.1 Composizione . . . . .	50
5.2.2 Aggiornamento Q-Table . . . . .	51
5.2.3 Modalità di utilizzo . . . . .	52
5.2.3.1 Prima modalità . . . . .	52
5.2.3.2 Seconda modalità . . . . .	52
5.2.4 Modifica Q-Table . . . . .	53
5.2.5 Riepilogo Q-Table . . . . .	54
5.3 Trade-off esplorazione o sfruttamento . . . . .	55
5.4 Decisione del policy maker . . . . .	55
5.4.1 Miglior azione all'istante $t$ . . . . .	55
5.4.2 Previsione del prossimo risultato . . . . .	55
5.4.3 Previsione risultati futuri . . . . .	56
5.4.4 Determinazione dell'azione ottimale . . . . .	57
5.5 Risultati . . . . .	58
5.5.1 Simulazioni a seme fisso . . . . .	58
5.5.2 Simulazioni a seme variabile . . . . .	60
5.6 Analisi di sensibilità . . . . .	61
5.6.1 Introduzione . . . . .	61
5.6.2 Costo $a_2$ ( $a_2\_cost\_1$ e $a_2\_cost\_2$ ) . . . . .	62
5.6.3 Costo $a_3$ . . . . .	64
5.6.4 Costo $a_1$ . . . . .	65
5.6.5 Confronto coppia di azioni $a_1, a_2$ con l'azione $a_3$ . . . . .	66
5.6.6 Costo degli infetti $\delta$ . . . . .	66
5.6.7 Parametri durata contagio $t_i, t_r$ . . . . .	68
<b>6 Conclusione</b>	<b>70</b>
<b>7 Appendice</b>	<b>75</b>
7.1 Parametri presenti nel modello . . . . .	75

# 1 Abstract

Sin dal principio la nostra specie ha cercato metodi per prendere decisioni riguardo a scelte presenti e future. Il processo decisionale comporta una serie di scelte e azioni da applicare che ogni individuo intraprende per un fine personale o collettivo. L'impiego di modelli matematici per analizzare il comportamento di sistemi dinamici risale all'inizio del XX secolo, quando la struttura matematica di tali modelli, inizialmente legata alla meccanica newtoniana, è stata generalizzata con successo per estenderne l'applicazione a contesti differenti [1].

In questa tesi si propone e si applica una metodologia per la ricerca della strategia ottimale a seconda del sistema e del suo stato in esame.

In primo luogo, vengono proposti i concetti alla base dell'Agent-Based Modelling (ABM), una metodologia di modellizzazione che consente di simulare sistemi complessi attraverso l'interazione di entità autonome denominate agenti. Successivamente, viene presentato il Reinforcement Learning (RL), una tecnica di apprendimento che consente l'identificazione della strategia ottimale per la risoluzione di un determinato problema.

Infine, viene presentata l'implementazione di un modello ABM progettato per simulare uno scenario sociale ed epidemiologico, al quale è applicata la tecnica di RL per identificare la strategia ottimale finalizzata alla minimizzazione del contagio. Il modello si baserà sul problema di El Farol Bar, estendendolo con una dinamica epidemiologica di tipo SIRS. In conclusione, saranno analizzati i risultati ottenuti dal modello e saranno esplorate possibili applicazioni gestionali per modelli di questa tipologia.

# Capitolo 1: Introduzione all'Agent Based Modelling e al Reinforcement Learning

## 2 ABM

### 2.1 Introduzione

L'Agent-Based Modelling, o modellazione ad agenti, permette la realizzazione di sistemi rappresentativi della realtà simulando di entità autonome denominate agenti.

Questo tipo di modellizzazione consente di simulare fenomeni complessi, come ad esempio la diffusione di una malattia infettiva, la formazione di un mercato, la diffusione di un'idea, ...

I fenomeni complessi vengono studiati dalla scienza dei sistemi complessi, la quale, rispetto ad altre discipline scientifiche, pone maggior enfasi sulle relazioni tra le componenti di un sistema [2]. Si definisce tanto più complesso un comportamento, tanto maggiore è la lunghezza della propria descrizione [3]. La lunghezza di una descrizione del comportamento di un particolare sistema dipende dal numero di possibili comportamenti che il sistema potrebbe esibire [4]. L'ABM abilita un approccio descrittivo dal basso verso l'alto consentendo l'implementazione di sistemi dalla prospettiva delle unità costituenti, possiamo quindi considerarlo come un mindset [5].

#### 2.1.1 Struttura

Un tipico modello ABM è composto dai seguenti tre elementi principali:

- **Agenti:** Sono le unità costituenti del modello, interagiscono tra loro e con l'ambiente circostante;
- **Regole:** Insieme di regole che governano il comportamento degli agenti;
- **Ambiente:** Spazio in cui e con cui gli agenti interagiscono.

### 2.1.2 Principi operativi dell'Agent-Based Modelling

Quando si esegue un modello ad agenti, essi ripetono i propri comportamenti frutto delle proprie regole o strategie, interagendo tra loro e con l'ambiente circostante.

Questo è il processo di simulazione alla base degli ABM, esso si lega profondamente agli elementi che ne compongono la struttura. Sebbene non sia una condizione necessaria, in genere questo processo opera lungo una linea temporale, come nelle simulazioni strutturate in time-stepped, activity-based o discrete-event [6]. Nello specifico:

- **Time-Stepped:** La simulazione procede in step temporali discreti, in cui gli agenti eseguono le loro azioni in sequenza durante ogni step  $t$ . Questo modello è particolarmente utile per la simulazione di sistemi in cui il tempo è un fattore critico;
- **Activity-Based:** I modelli activity-based si concentrano maggiormente su quali attività vengono svolte dagli agenti, sulla loro durata, sul momento e luogo in cui si svolgono, sui soggetti coinvolti e sulle scelte che gli agenti compiono per portarle a termine. Questo modello può essere particolarmente utile per la valutazione dei risultati ottenuti a seguito dell'applicazione di determinate politiche sulla popolazione [7];
- **Discrete-Event:** DES (Discrete Event Simulation) consente la rappresentazione di sistemi come sequenza di eventi che possono verificarsi in momenti specifici. Solo eventi possono modificare il sistema, si assume quindi che tra un evento e l'altro il sistema rimanga invariato. Ciò permette al modello di passare all'istante di tempo  $t$  del prossimo evento, questo processo viene definito come "next-event time progression".

**Esempio 1.** *Ipotizziamo che ad un distributore automatico si affacci un agente denominato cliente, egli può acquistare un prodotto modificando lo stato del sistema distributore. Una volta che l'agente si allontana dal distributore, fintanto che non si affaccerà un nuovo cliente gli istanti  $t$  sono irrilevanti, perciò vengono ignorati.*

## 2.2 Agenti

Gli agenti, fondamenta di un ABM, possono rappresentare persone, animali o entità astratte. Essi interagiscono sia tra loro che con l'ambiente circostante. Ogni agente dispone di

una capacità decisionale autonoma e agisce in base a un insieme di regole predefinite.

Sebbene il comportamento degli agenti faccia affidamento a un set di regole determinate, questo insieme non è fisso e il loro comportamento può essere modificato a seguito di interazioni tra gli stessi o con l'ambiente. L'Agent-based modelling offre un modo per simulare sistemi sociali composti da agenti che interagiscono e si influenzano a vicenda, imparando dalle esperienze effettuate e adattando i propri comportamenti in modo che siano più consoni sistema di cui fanno parte [6]. Ogni agente possiede un insieme di caratteristiche: alcune, come quelle comportamentali, possono variare nel tempo, mentre altre, come quelle genetiche, rimangono invariate [8].

Gli agenti possono essere modellati per intraprendere azioni pertinenti al sistema che si vuole rappresentare, più agiscono coerentemente con il sistema che rappresentano, più il modello risulta rappresentativo della realtà.

La caratteristica distintiva degli agenti in un ABM è proprio l'autonomia, ovvero la capacità di agire senza guide esterne rispetto alle situazioni che egli incontra nel sistema. Gli agenti sono dotati di comportamenti che permettono loro di prendere decisioni autonome [6] e di conseguenza intraprendere determinate azioni.

La mancata presenza di guide esterne che determinano le decisioni degli agenti permette di considerare i modelli ad agenti come sistemi decentralizzati, senza un'autorità centrale che determini le azioni degli agenti. "Gli agenti ottengono informazioni tramite interazioni con altre unità costituenti (non qualsiasi agente o tutti gli agenti) e dall'ambiente localizzato (non da nessuna parte dell'intero ambiente)" [6].

### **2.2.1 Fenomeno emergente**

Un elemento caratteristico dell'ABM è la rilevanza data all'eterogeneità degli agenti, ovvero alla diversità di comportamenti, regole e caratteristiche degli agenti appartenenti ad un sistema. Essendo l'ABM un approccio bottom-up è possibile differenziare gli enti costituenti, mentre utilizzando altre tecniche per la modellizzazione di sistemi complessi questa distinzione risulterebbe più complicata. L'approccio dal basso verso l'alto consente la rappresentazione di comportamenti eterogenei degli agenti e porta ad emergere fenomeni complessi a livello di sistema [3].

### 2.2.2 Caratteristiche degli agenti

Le caratteristiche degli agenti possono essere molteplici, le principali sono:

- **Unicità:** L'agente è modulare, unico e identificabile come individuo. Ogni agente possiede un insieme di attributi che lo rendono unico e distinguibile dagli altri agenti [6];
- **Autonomia:** L'agente è autonomo, può interagire con altri agenti e con l'ambiente circostante senza una guida esterna [9];
- **Socialità:** L'agente è sociale e il suo comportamento è influenzato dalle interazioni dinamiche con altri agenti. In base alla composizione del modello potrebbero essere presenti algoritmi o protocolli per la comunicazione, diffusione, scambio d'informazioni e movimentazione degli agenti;
- **Adattabilità:** L'agente può avere la capacità di apprendere e adattare i propri comportamenti in base alle esperienze passate.

Altre caratteristiche, non sempre presenti in un modello ad agenti, vengono descritte di seguito:

- **Automa:** L'agente possiede uno stato, definito dalle proprie variabili. Il modello ad agenti è caratterizzato da uno stato complessivo che include sia lo stato individuale di ciascun agente sia lo stato dell'ambiente in cui questi operano. Il comportamento dell'agente dipende dallo stato in cui si trova: Quanto più complesso e articolato sarà il suo stato (anche in relazione al numero di variabili), tanto più variegati saranno i suoi comportamenti. Si dice che un agente possieda uno stato quando vi è una condizione definita, riconoscibile e ricorsiva [1]. Non tutti gli agenti possono essere quindi rappresentati con un automa.
- **Percezione:** Capacità di raccogliere stimoli dall'ambiente interagendo con esso [1].
- **Proattività:** Possibilità di non agire rispondendo solo agli stimoli dell'ambiente circostante, ma perseguitando obiettivi personali [10].
- **Scopo:** Capacità di raggiungere un obiettivo definito agendo in modo proattivo e reattivo [1].

## 2.3 Vantaggi

I modelli ABM permettono di:

- **Cogliere fenomeni emergenti:** Negli ABM è possibile identificare con facilità fenomeni emergenti, ovvero comportamenti del sistema non direttamente derivabili dalle regole che governano il comportamento degli agenti. Questi fenomeni non sono noti a priori, poiché emergono in seguito alla descrizione primaria delle unità costituenti negli ABM. I fenomeni emergenti sono parte della complessità che emerge dalle azioni individuali e dalle interazioni tra gli agenti;
- **Produrre la più naturale descrizione di un sistema:** Gli ABM sono i modelli più naturali per descrivere un sistema complesso composto da entità "comportamentali" [5]. Ad esempio risulta più semplice descrivere un mercato come un insieme di agenti che interagiscono tra loro (punto di vista delle unità costituenti), piuttosto che come un insieme di equazioni differenziali (punto di vista globale). Questo fenomeno è accentuato quando:
  - Il comportamento degli agenti non è lineare e non può essere descritto da tassi;
  - Il comportamento degli agenti è complesso. Sebbene sia possibile l'utilizzo di equazioni, le equazioni differenziali complicherebbero esponenzialmente il modello;
  - Le attività sono metodi più naturali per descrivere il sistema rispetto che processi;
  - I comportamenti degli agenti sono stocastici. Negli ABM le componenti randomiche vengono applicate in posizioni specifiche, piuttosto che un termine numerico aggiunto in un'equazione aggregata.
- **Adattarsi:** La flessibilità degli ABM spazia in varie dimensioni. Essi possono essere utilizzati in vari contesti, e facilmente modificati cambiando i parametri del modello. Ad esempio si può osservare un fenomeno su una porzione limitata della popolazione e poi scalare il modello ad una porzione più ampia, verificando se il fenomeno si ripresenta.  
Sebbene ogni modello abbia parametri specifici, e possibili interdipendenze tra parametri, generici parametri chiave per scalare il modello sono:

- $n$ : Il numero di agenti consente di scalare il modello ad una popolazione più ampia;
- $T$ : Il tempo di simulazione, ovvero il numero di iterazioni, consente di scalare il modello ad un periodo più lungo;

## 2.4 Svantaggi

Sono note alcune problematiche legate alla modellizzazione ad agenti. In primis deve avere uno scopo ed essere costruito con il giusto livello di dettaglio per rappresentare il sistema, i modelli "general-purpose" tendono ad essere meno efficaci. Questa problematica è presente anche in altre tecniche di modellizzazione, nel caso dell'ABM rende la tecnica un'arte più che una scienza [5].

In molti casi gli agenti sono esseri umani, quindi dotati di potenziali comportamenti irrazionali, scelte soggettive e psicologie complesse. Questo grado di dettaglio rende difficile la comprensione del modello anche ad esperti del settore.

Infine, per definizione, i modelli ABM analizzano un sistema partendo dal punto di vista delle unità costituenti (gli agenti). Simulare un numero elevato di agenti in più iterazioni richiede un'elevata potenza computazionale.

Per ovviare a quest'ultima problematica è possibile utilizzare tecniche ibride. Esse combinano l'ABM con altre tecniche come l'EBM (Equation Based Modelling), "iniziano la simulazione come ABM e passando a EBM dopo che il numero di individui infetti è abbastanza grande da supportare un approccio medio della popolazione" [11].

## 2.5 Metodi per la descrizione

Come descritto nella sezione relativa agli svantaggi degli ABM, uno di essi è proprio la complessità descrittiva che deriva dall'approccio bottom-up. La tecnica ABM spesso produce modelli particolarmente complessi, che risultano difficili da comprendere e comunicare ad altri ricercatori.

Per ovviare a questa problematica si sono cercati strumenti per la descrizione degli ABM. Uno di questi è l'ODD (Overview, Design concepts, Details) realizzato da Grimm et al in [12].

### 2.5.1 Introduzione ODD

ODD nasce a seguito dei problemi identificati nella descrizione di modelli ABM, in quanto prima di esso non esisteva uno standard per la descrizione e molto spesso venivano descritti verbalmente senza una chiara indicazione delle equazioni, regole e processi utilizzati.

Il metodo ODD si propone di identificare un protocollo strutturato per la descrizione di ABM, questo strumento si compone di una sequenza ben precisa.

### 2.5.2 Composizione ODD

Panoramica	Scopo
	Variabili di stato
	Descrizione dei processi e composizione
Concetti di design	Concetti di design
Dettagli	Inizializzazione
	Input
	Sotto modelli

Tabella 1: Composizione ODD

Come descritto anche in Tabella 1, il protocollo ODD si compone di tre sezioni principali: Panoramica, Concetti di design e Dettagli. Queste tre macro categorie si compongono di altre sette micro-sezioni, più precisamente:

- **Panoramica:** Permette una panoramica generale dello scopo e struttura del modello. Dopo aver letto la panoramica dovrebbe essere possibile scrivere lo scheletro del modello in qualsiasi linguaggio di programmazione ad oggetti. Lo scheletro include gli oggetti e le entità (agenti e ambienti);
  - **Scopo:** Contiene un'introduzione al modello e lo scopo della simulazione;
  - **Variabili di stato:** In questa sezione si trova l'intero set di variabili di stato che definiscono il modello. Il termine variabile di stato si riferisce alle variabili che caratterizzano le entità di basso livello, come ad esempio gli agenti. È importante non confondere le variabili di stato con le variabili ausiliarie o globali.

Le variabili globali contengono informazioni che possono essere dedotte dalle unità di basso livello e le loro entità.

**Esempio 2.** *Ipotizzando un modello ABM per la simulazione di una popolazione in una città, le variabili di stato degli individui potrebbero essere: età, sesso, stato di salute, residenza, ... Le variabili ausiliarie potrebbero essere il numero di individui, numero di individui infetti, densità di individui residenti in un'area, ... Conoscendo la variabile di stato "residenza", è possibile calcolare la variabile globale "densità di individui residenti in un'area".*

- **Descrizione dei processi e composizione:** In questa sezione vengono descritti i processi implementati nell'ABM, la cui conoscenza è fondamentale per comprenderlo. In molti casi questa sezione potrebbe essere accompagnata da diagrammi di flusso per facilitarne la comprensione;
- **Concetti di design:** Descrive i concetti generali dietro al modello. Lo scopo di questa sezione è collegare il design del modello ai fenomeni derivanti dal sistema complesso (fenomeni emergenti). Per la redazione di questa sezione, si farà riferimento ai seguenti punti che rispondono alle domande più comuni legate ai modelli ABM:
  - **Fenomeni emergenti:** Quali fenomeni emergenti fuoriescono dalle dinamiche degli agenti?
  - **Adattamento:** Come gli agenti si adattano a nuove condizioni dell'ambiente o della società?
  - **Ricerca dell'ottimo:** La ricerca dell'ottimo è modellata esplicitamente o implicitamente? Se esplicitamente, come gli agenti cercano l'ottimo? Se implicitamente, quali sono le forze che spingono gli agenti verso l'ottimo?
  - **Previsione :** Nello stimare le conseguenze future delle proprie azioni, come gli agenti predicono il futuro?
  - **Sensibilità:** Quali variabili di stato conoscono gli agenti e come queste influenzano le loro decisioni?
  - **Interazioni:** Quale tipo di interazioni esistono tra gli agenti e l'ambiente?
  - **Stocasticità:** La stocasticità fa parte del modello?
  - **Collettività:** Sono presenti gruppi sociali tra gli agenti?

- **Osservazione:** Come vengono raccolti i dati dal modello?
- **Dettagli:** In questa sezione vengono inseriti i dettagli omessi nella panoramica;
  - **Inizializzazione:** Si descrive come vengono inizializzati gli agenti e l'ambiente ad inizio simulazione;
  - **Input:** Le dinamiche ambientali possono variare nel tempo, gli agenti rispondono a tali dinamiche denominate input. L'output del modello risponde agli input forniti, in questa sezione vengono definiti gli input del modello;
  - **Sotto modelli:** Questa si considera un'estensione della sezione "Descrizione dei processi e composizione" nel quale vengono descritti i sotto modelli che rappresentano i processi già citati, includendo anche parametri, equazioni, regole e assunzioni dietro regole ed equazioni;

## 3 Reinforcement Learning

### 3.1 Introduzione

Il Reinforcement Learning implementa agli agenti o soggetti della simulazione la capacità di affrontare un problema, imparando attraverso interazioni trial-and-error con un ambiente dinamico [13]. Questa tecnica rende il processo decisionale sequenziale poiché l'agente giunge alla decisione finale dopo diverse interazioni con l'ambiente. Si tratta, quindi, di un processo di decisione in cui l'ambiente (spesso stocastico) evolve secondo dinamiche di transizione che dipendono dalla storia del sistema e dalle azioni (decisioni) intraprese da un decisore autonomo [14]. L'obiettivo del processo decisionale sequenziale è quello di intraprendere azioni in modo da massimizzare una certa misura dell'utilità cumulativa prevista. In quanto il reinforcement learning può essere applicato in vari ambiti e discipline per affrontare problemi decisionali complessi, in contesti medici l'utilità può per esempio corrispondere ad una misura composita del benessere del paziente oppure al numero di infezioni nel contesto del controllo della diffusione di una malattia infettiva [15]. Nelle seguenti sezioni verranno descritti i principi del Reinforcement learning, facendo anche riferimento all'applicazione in simulazioni ABM.

### 3.2 Modello

In un classico modello di apprendimento per rinforzo, un agente è collegato al suo ambiente tramite percezione ed azione [13]. Ogni step ( $t$ ) l'agente riceve un input ( $i$ ) ed indicazioni riguardanti lo stato in cui si trova ( $s$ ). L'agente o il soggetto della simulazione prende quindi una decisione sull'azione da prendere ( $a$ ) che cambierà lo stato dell'ambiente. La transizione tra due stati dell'ambiente genera un risultato ( $r$ ) che può essere positivo o negativo in relazione allo spettro dei risultati.

Riassumendo, in un modello di reinforcement learning sono presenti i seguenti elementi:

- $t$  : Istante di tempo;
- $i$  : Input che l'agente riceve dal sistema;
- $s$  : Indicazioni che l'agente riceve dal sistema riguardo lo stato in cui si trova;
- $a$  : Azione intrapresa dall'agente ad ogni istante  $t$  a seguito di valutazioni o strategie su  $i$  e  $s$ ;

- $r$  : Risultato restituito dal sistema generato dal cambio di  $s$  causato da  $a$ .

### 3.3 Q-Table

La q-table rappresenta uno strumento per il salvataggio dei risultati ( $r$ ) ottenuti dalle azioni intraprese dall'agente ( $a$ ). Essa è composta da:

- $S$  : Un discreto set di stati dell'ambiente;
- $A$  : Un discreto set di azioni dell'agente.

Inizialmente la q-table viene generata con valori nulli che verranno in seguito sovrascritti con valori ottenuti dai risultati ( $r$ ). La Q-table presenta dimensioni pari a  $S \cdot A$ .

**Esempio 3.** Considerando l'esempio di un lago ghiacciato diviso in 16 quadrati (matrice  $4 \cdot 4$ ), dove l'agente dal punto A deve raggiungere il punto B, evitando buche e percorrendo la strada più sicura e veloce.

In ogni stato l'agente deve scegliere una tra 4 azioni:

1. *Su* : Procedere nel blocco sopra la sua posizione;
2. *Giù* : Procedere nel blocco sotto la sua posizione;
3. *Destra* : Procedere nel blocco a destra;
4. *Sinistra* : Procedere nel blocco a sinistra.

In questo caso la q-table sarà una matrice  $16 \cdot 4$  contenente 64 valori ( $16 \cdot 4 = 64$ ), in quanto 16 sono i possibili stati ( $S$ ), mentre 4 le possibili azioni ( $A$ ).

Ad ogni istante  $t$ , la q-table, verrà aggiornata il valore di risultato  $r_t$  (affronteremo il processo di aggiornamento nella prossima sezione) nella posizione corrispondente allo stato  $s_t$  (stato del sistema in cui è presente l'agente) e all'azione  $a_t$  (azione intrapresa dall'agente).

La q-table viene quindi aggiornata nella posizione:  $Q(s_t, a_t)$ .

Con l'aumentare di  $t$ , si otterrà un q-table sempre meno popolata da valori nulli, bensì incrementeranno i valori generati ( $r$ ) che mapperanno ed istruiranno l'agente in merito agli stati in cui si è già trovato e le azioni, vincenti o evitabili, precedentemente intraprese.

### 3.4 Ricompensa e aggiornamento q-table

La ricompensa è il valore che indica la bontà della scelta effettuata dall'agente a seguito della decisione sulla strategia o azione ( $a$ ) da perseguire.

In certi casi la ricompensa può essere anche nulla o negativa, rispettivamente lasciando invariata al q-table o identificando azioni  $a$  da non intraprendere in determinati stati  $s$ .

**Esempio 4.** Riprendendo il precedente esempio del lago ghiacciato (Esempio 3), potremmo attribuire un risultato positivo alle coppie stato-azione corrispondenti ai blocchi percorsi dall'agente quando riesce a raggiungere il punto B con successo. Potremmo invece salvare un risultato negativo in corrispondenza delle coppie stato-azione che porteranno l'agente a cadere in una buca.

Definito il risultato dell'azione  $a$ , esso viene salvato nella q-table. Il salvataggio nella q-table viene effettuato per mezzo di equazioni apposite, una di queste è la "Bellman Optimality Equation" che conserva in parte il risultato precedentemente salvato nella q-table.

Viene definita come:

$$Q_{new}(s_t, a_t) = Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot \max_{a'} Q(s_t, a') - Q(s_t, a_t))$$

Dove:

- $Q(s_t, a_t)$  : Rappresenta il valore salvato attualmente in q-table in corrispondenza dello stato  $s_t$  e l'azione  $a_t$ .  $Q_{new}(s_t, a_t)$  sarà quindi il valore con cui verrà sostituito  $Q(s_t, a_t)$ .
- $\alpha$  : Identifica il coefficiente di apprendimento (learning rate). Ammette valori tra 0 e 1 e determina, data una coppia stato azione, quanta informazione già salvata nella q-table portare a nuovo nel futuro valore salvato in q-table. Determina quindi l'incidenza del valore attuale, in corrispondenza della coppia stato valore ( $Q(s_t, a_t)$ ), sul nuovo valore inserito in q-table ( $Q_{new}(s_t, a_t)$ ). Con  $\alpha = 0$  il valore in q-table non cambierebbe mai, mentre con  $\alpha \rightarrow 1$  cambierebbe molto velocemente senza considerare i valori precedentemente salvati.
- $\gamma$  : Si tratta del fattore di sconto (discount factor). Ammette valori tra 0 e 1 ed è una misura di quanto l'agente consideri ricompense future rispetto a ricompense

imminenti. Con  $\gamma \rightarrow 0$  l'agente preferirà ricompense nel breve periodo, mentre con  $\gamma \rightarrow 1$  ricompense future.

- $(r_t + \gamma \cdot \max_a(Q(s_{t+1}, a_t)) - Q(s_t, a_t))$  : Questa parte dell'equazione rappresenta il nuovo valore appreso, che verrà aggiunto ad il valore già conosciuto.
- $\max_a(Q(s_{t+1}, a_t))$  : Valore in corrispondenza della coppia stato parsi a  $t + 1$ , ovvero lo stato in cui l'agente si troverà nel prossimo istante di tempo decisionale, ed in corrispondenza della miglior azione, ovvero colei con valore maggiore (o valore minore se si considerano costi).

**Esempio 5.** Seguendo l'esempio del lago ghiacciato (Esempio 3) potremmo utilizzare la Bellman Optimality Equation per l'aggiornamento dei dati in q-table, definendo quindi i parametri come:

- $\alpha = 0.5$  : In questo modo si avrebbe un perfetto bilanciamento tra valori derivanti dalle precedenti esperienze dell'agente e valori identificati ad ogni nuovo istante.
- $\gamma = 0.25$  : Così facendo l'agente preferirà ricompense nel breve periodo, limitando il numero di errori e la probabilità di cadere in buche.

### 3.5 Esplorazione o Sfruttamento (Exploration or Exploitation)

In una simulazione, ad ogni ciclo, il soggetto della simulazione sceglie se esplorare nuove strategie o sfruttare quelle già conosciute. La scelta tra esplorazione (Exploration) e sfruttamento dei dati già raccolti (Exploitation) rappresenta uno dei principali trade-off del reinforcement learning.

#### 3.5.1 Esplorazione

Si tratta della fase di esplorazione di nuove strategie. Esplorando l'agente aumenta la probabilità di identificare strategie migliori. Una maggiore esplorazione aumenta quindi la probabilità di trovare punti di massimo globali, aumentando però di contro le possibilità di incumberci in risultati negativi, che impattino nel risultato complessivo della simulazione. Ad esempio, nelle simulazioni ad agenti (ABM) viene definita una soglia di esplorazione, in base alla quale l'agente sperimenta strategie differenti da quelle ottimali già esplorate e salvate nella Q-table, qualora la condizione associata alla soglia venga soddisfatta.

Identifichiamo il threshold di esplorazione (explor rate) con  $\varepsilon$  e rappresenta la probabilità di esplorazione ad ogni ciclo.

### 3.5.2 Sfruttamento

Lo sfruttamento consente di limitare il rischio di cicli con risultato negativo, in quanto viene scelta l'azione in base alla migliore coppia stato-valore presente nella q-table. Essere eccessivamente avidi (Greedy) e limitare troppo la fase di esplorazione, potrebbe non permettere l'ottenimento della massima ricompensa, portando ad un comportamento sub-ottimale, limitando quindi la simulazione a risultati di massimo locali.

### 3.5.3 Trade-off

Per definire e trovare la strategia ottima, è essenziale esplorare nuove strategie che possano risultare più efficaci di quelle salvate nella q-table. Aumentando il numero di esplorazioni, si avrà come effetto anche una riduzione della ricompensa nel risultato finale, in quanto così facendo aumenterebbero anche i cicli con risultati negativi.

Nel caso di un ABM, quando un agente esplora, ottiene stime più accurate dei valori di azione, mentre quando sfrutta quanto ha già appreso, potrebbe ottenere una maggior ricompensa. L'agente non può, tuttavia, scegliere di fare entrambe le cose contemporaneamente, ciò viene chiamato dilemma esplorazione-sfruttamento (trade-off). La scelta ricade quindi tra l'esplorazione, lo sfruttamento o strategie miste.

## 3.6 Algoritmo di Epsilon-Greedy

Una soluzione al trade-off tra esplorazione e sfruttamento è l'algoritmo di Epsilon-Greedy, ovvero un metodo semplice per bilanciare l'esplorazione e lo sfruttamento scegliendo tra esplorazione e sfruttamento in modo parzialmente stocastico. L'epsilon-greedy, dove  $\varepsilon$  si riferisce alla probabilità di scegliere di esplorare. In genere si parte da una prima fase di esplorazione, per poi passare ad una fase di sfruttamento, in modo da ottenere risultati migliori.

$$Azione_t = \begin{cases} \max Q_t(a) & \text{probabilità } 1 - \varepsilon \\ \text{azione random } (a) & \text{probabilità } \varepsilon \end{cases}$$

---

**Algoritmo 1** Pseudo codice

---

```
p = random()  
if p ≥ ε then  
    a = azione random  
else  
    a = max Qt(a)  
end if
```

---

**Esempio 6.** Nel caso di un'azienda di logistica, immaginiamo di avere ogni giorno  $n$  automezzi che consegnano merce presso la locazione dei clienti, e che l'obiettivo sia quello di ridurre la strada percorsa da ogni singolo automezzo in modo da risparmiare in carburante e in costi di manutenzione.

Ipotizzando che su 100 giorni gli automezzi abbiano sempre lo stesso punto di partenza e di arrivo, immaginiamo che ogni giorno l'autista si trovi di fronte alla scelta di esplorare una nuova strada o percorrerne una già esplorata. Maggiore sarà l'explore rate  $ε$ , maggiore sarà la probabilità di consegne effettuate in esplorazione consumando un maggior numero di risorse. Aumentando il numero di consegne in esplorazione aumenterà anche la probabilità di trovare una strada ancor più veloce, che faccia risparmiare denaro per la consegna e le successive.

Vogliamo esplorare l'ambiente il più possibile, sebbene diventi ogni ciclo meno interessante, poiché si conoscono sempre più coppie stato-azione [16].

L'algoritmo Epsilon-Greedy definisce una riduzione dell'explore rate  $ε$  ogni ciclo o  $n$  cicli, di un valore scelto a priori ( $η$ ).

Il threshold di esplorazione diminuirà fino al raggiungimento di un valore pari a 0, momento in cui ipotizziamo la simulazione sufficientemente istruita per basare le future scelte esclusivamente su quanto già esplorato e quindi sulle coppie stato-azione salvate nella q-table. In questo modo aumenta la probabilità di effettuare scelte che portino a risultati complessivamente migliori, in quanto quando l'agente è non è ancora istruito e preferirà l'esplorazione, mentre con l'aumentare di  $t$ , quando la q-table sarà sufficientemente popolata utilizzerà le coppie stato-azione. Calcoliamo quindi l'explore rate di ogni ciclo come:

$$ε_t = ε_{t-1} - η$$

**Esempio 7.** Facendo riferimento al precedente esempio riguardante l'azienda logistica (Esempio 6), immaginiamo di applicare l'algoritmo Epsilon-Greedy. In questo caso, ogni giorno, l'autista avrà la probabilità di esplorare una nuova strada pari ad un valore  $\varepsilon_t = \varepsilon_{t-1} - \eta$ . Intuiamo che ad un certo istante  $t$ ,  $\varepsilon$  raggiungerà un valore pari a 0, e che quindi di conseguenza la probabilità che l'autista scelga di esplorare una nuova strada sarà nulla.

# Capitolo 2: Implementazione di ABM e Reinforcement Learning

## 4 ABM Epidemiologico

### 4.1 Introduzione

In questa sezione viene discussa l'applicazione di un modello ABM per la simulazione di una epidemia. Il modello ad agenti implementa in primis il problema di El Farol Bar, introducendo poi un'epidemia che si diffonde tra gli agenti. Infine, vengono prese in esame decisioni relative alla riduzione del contagio mediante l'introduzione di un nuovo agente, il policy maker.

### 4.2 Panoramica

#### 4.2.1 Scopo

Lo scopo è quello di esplorare le relazioni tra le decisioni sociali e le dinamiche epidemiologiche. Il modello è un'estensione del problema di El Farol Bar e mira a conoscere ed identificare fenomeni emergenti nel sistema, difronte ad una minaccia epidemiologica.

#### 4.2.1.1 El Farol Bar

Il problema del bar di El Farol, proposto dall'economista Brian W. Arthur nel 1994 in [17], è un problema di allocazione vincolata composto da agenti non cooperativi e appartiene alla teoria dei giochi. Nel sistema un numero di  $n$  agenti deve decidere ogni istante  $t$  se andare al bar o rimanere a casa. Se il numero di agenti che va al bar è maggiore di un threshold  $t_e$ , il bar sarà troppo affollato e nessuno riceverà ricompensa. Ciò implica che nessun agente sarà effettivamente contento di essersi recato al bar in quanto troppo affollato. Se invece il numero di agenti che va al bar è minore di  $t_e$ , allora i presenti tutti riceveranno una ricompensa e saranno felici del tempo passato al bar. Questo problema analizza le modalità con cui è possibile raggiungere un ottimo collettivo in un contesto di allocazione delle risorse e in una situazione in cui l'adozione della medesima strategia da parte di tutti gli agenti condurrebbe a un fallimento collettivo. [18]

Il principale vincolo del problema è che gli agenti non possono comunicare, decidere coordinatamente o confrontarsi riguardo alla decisione relativa alla presenza al bar. Alcun agente ha modo di conoscere a priori il grado riempimento del bar in un'istante  $t + 1$  nel futuro, se non presentandosi e potenzialmente rimanendo deluso.

Senza potersi confrontare, gli agenti possono superare questo vincolo affidandosi a supposizioni o strategie personali, che potrebbero essere basate su esperienze passate o strategie di gioco.

Se tutti gli agenti seguissero la medesima strategia, allora si garantirebbe il fallimento del sistema. Si avrebbe infatti uno scenario in cui si alternerebbero istanti in cui tutti gli agenti si recherebbero al bar, rimanendo così scontenti, ed istanti in cui nessuno si recherebbe al bar, dove il grado di riempimento del bar sarebbe sempre al di sotto di  $t_e$  ma nessun agente ne trarrebbe mai vantaggio.

Esiste un'unica strategia mista simmetrica di equilibrio di Nash, in cui ciascun agente decide di andare al bar con una certa probabilità. Questa probabilità è calcolata tenendo conto del numero di agenti, della soglia di affollamento del bar e dell'utilità relativa derivante dal frequentare un bar affollato o poco affollato rispetto a restare a casa.

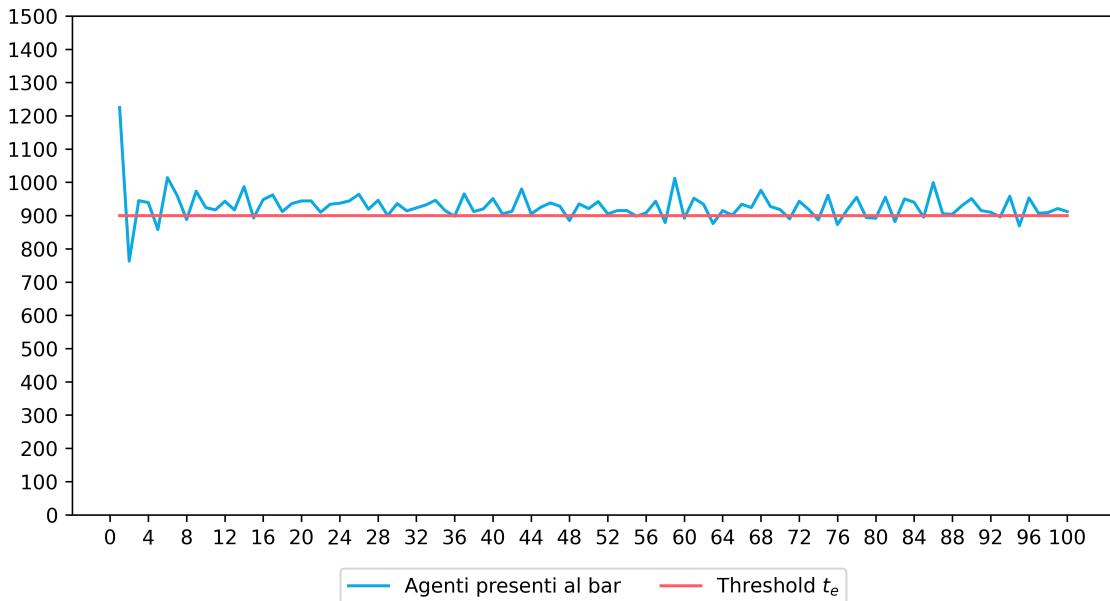


Figura 1: Simulazione default El Farol ( $seed = 59296$ )

Come è possibile notare nella Figura 1, il numero di agenti che si reca al bar varia in

ogni istante  $t$ , ciò implica che gli agenti del modello implementato non seguono la stessa strategia. Il numero di agenti che si reca al bar è rappresentato dalla linea blu, mentre il threshold  $t_e$  è raffigurato dalla linea rossa. Il funzionamento del modello implementato è in linea con quanto detto da Brian W. Arthur nel 1994 in [17], infatti il numero di agenti sembra oscillare attorno al threshold  $t_e$ .

L'utilizzo dell'ABM per la modellazione di un problema di teoria dei giochi come El Farol Bar permette di studiare le dinamiche sociali, le strategie adottate dagli agenti ed i loro comportamenti.

#### 4.2.1.2 Analisi di sensibilità El Farol

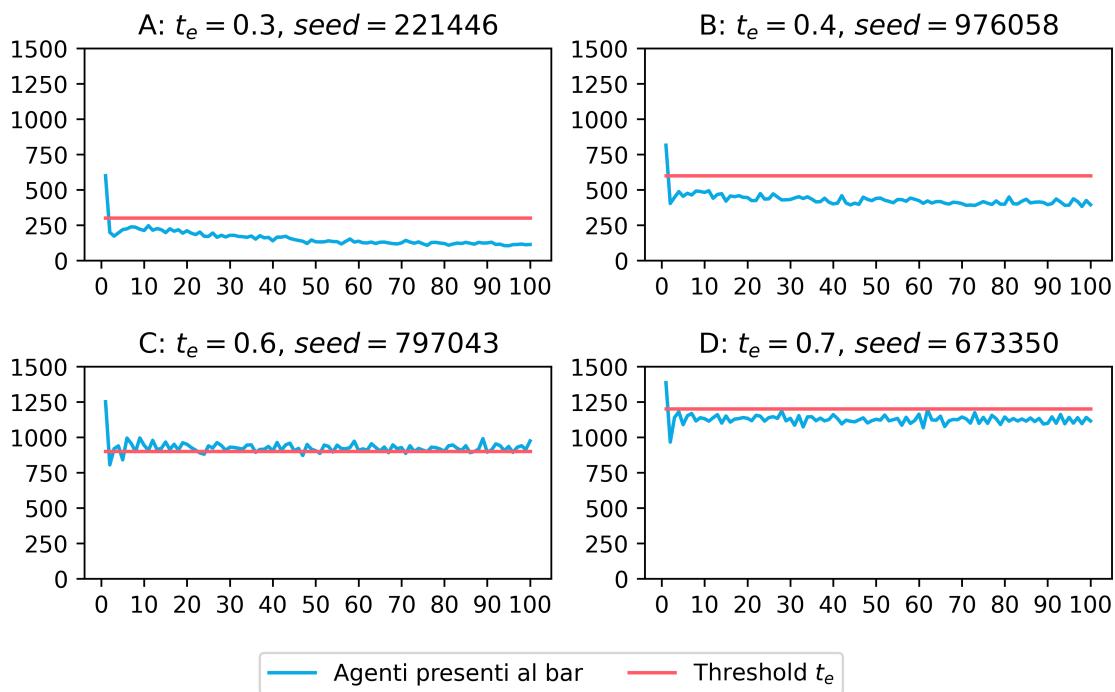


Figura 2: Analisi sensibilità El Farol

Nella Figura 2 è possibile osservare l'andamento del numero di agenti che si reca al bar in relazione al threshold  $t_e$ . Come è possibile notare, il numero di agenti che si reca al bar è proporzionale al threshold  $t_e$ . Aumentando il threshold  $t_e$ , il numero di agenti che si reca al bar aumenta, mentre diminuendo il threshold  $t_e$  il numero di agenti che si reca al bar diminuisce.

$t_e$  rappresenta la capacità massima del bar per mantenere la soddisfazione degli avventori. Più alto è questo valore, maggiore sarà il numero di persone che potranno frequentare

il bar senza influenzare negativamente il livello di felicità complessivo dei presenti.

Come notato anche da Brian W. Arthur nel 1994 in [17], con un threshold  $t_e = 0.6$ , gli agenti si auto-organizzano in modo che la presenza media degli agenti presenti al bar sia 900 ( $t_e \cdot c_b = 0.6 \cdot 1500 = 900 = E(Attendance)$ ), dove in questo caso 1500 non è il di agenti  $n$  ma la capacità massima del bar  $c_b$ ). Una spiegazione di questo fenomeno considera il valore  $t_e = 0.6$  come attrattore naturale del problema; infatti, se lo si vede come un puro gioco di previsione, una strategia mista di previsione superiore a 900 con probabilità 0,4 e sotto di essa con probabilità 0,6 è un equilibrio di Nash.

#### 4.2.1.3 Estensione epidemiologica

Il modello ABM implementato estende il problema di El Farol Bar introducendo un'epidemia. Gli agenti che si recano al bar possono infettarsi e diventare contagiosi. Gli agenti infetti possono a loro volta infettare altri agenti in base al loro livello di contagio  $c_a$ . Ogni istante  $t$  gli agenti possono diventare infetti, suscettibili o recuperati. Gli agenti infetti possono infettare altri agenti suscettibili, mentre gli agenti recuperati non possono più essere infettati. L'agente può quindi trovarsi nei seguenti stati:

- $A_S$  (**agente suscettibile**): In questo caso l'agente non è infetto ma può essere contagiato se si reca al bar.
- $A_I$  (**agente infetto**): L'agente è infetto e recandosi al bar può infettare un numero  $n$  di agenti in base al proprio livello di contagio  $c_a$ . Non è detto che l'agente possa sempre presentarsi al bar se infetto, l'ipotesi è che se l'agente accusi sintomi troppo gravi non si presenterà al bar.
- $A_R$  (**agente recuperato**): L'agente è guarito e non può essere infettato. L'ipotesi è che abbia sviluppato gli anticorpi e non possa più essere contagiato per un determinato periodo di tempo (sia esso anche infinito).

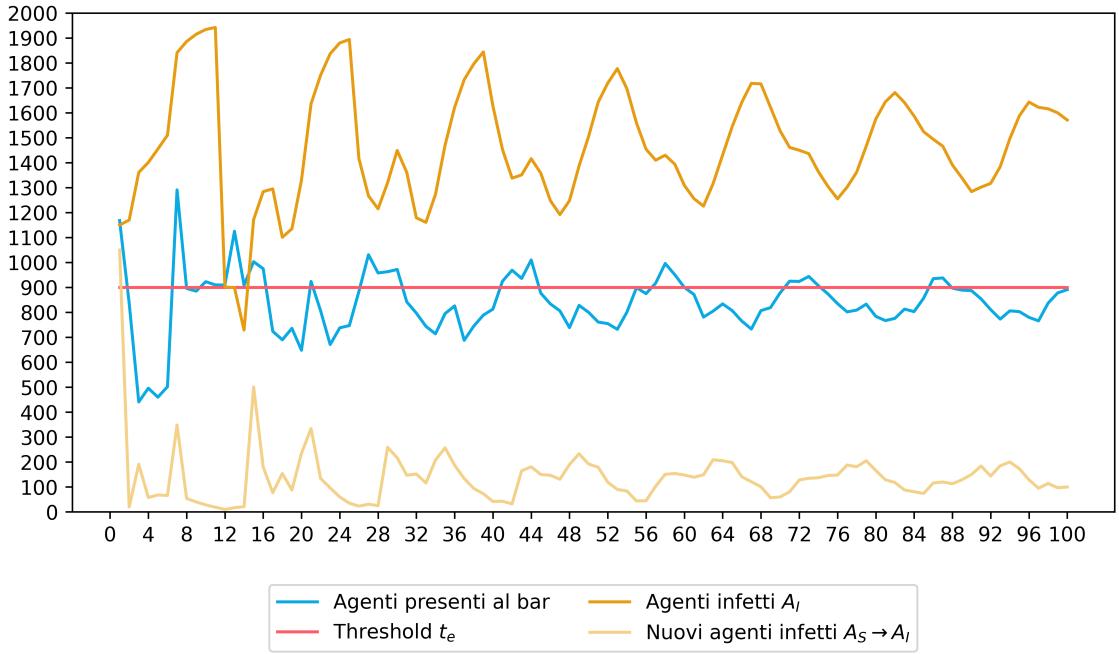


Figura 3: Simulazione default epidemia ( $seed = 808698$ )

In Figura 3 è possibile osservare il comportamento degli agenti in un modello ABM che implementa il problema di El Farol Bar con l'aggiunta di un'epidemia. In questo caso si può notare come l'epidemia sia particolarmente aggressiva, il numero di infetti cresce fino a raggiungere un intorno che va dai 917 ai 1840 agenti entro la decima iterazione  $t$ .

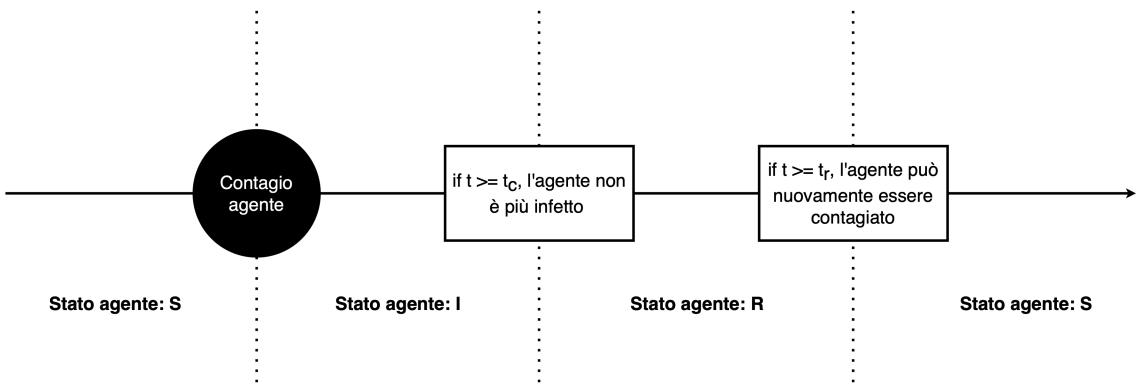


Figura 4: Diagramma di flusso epidemia SIRS

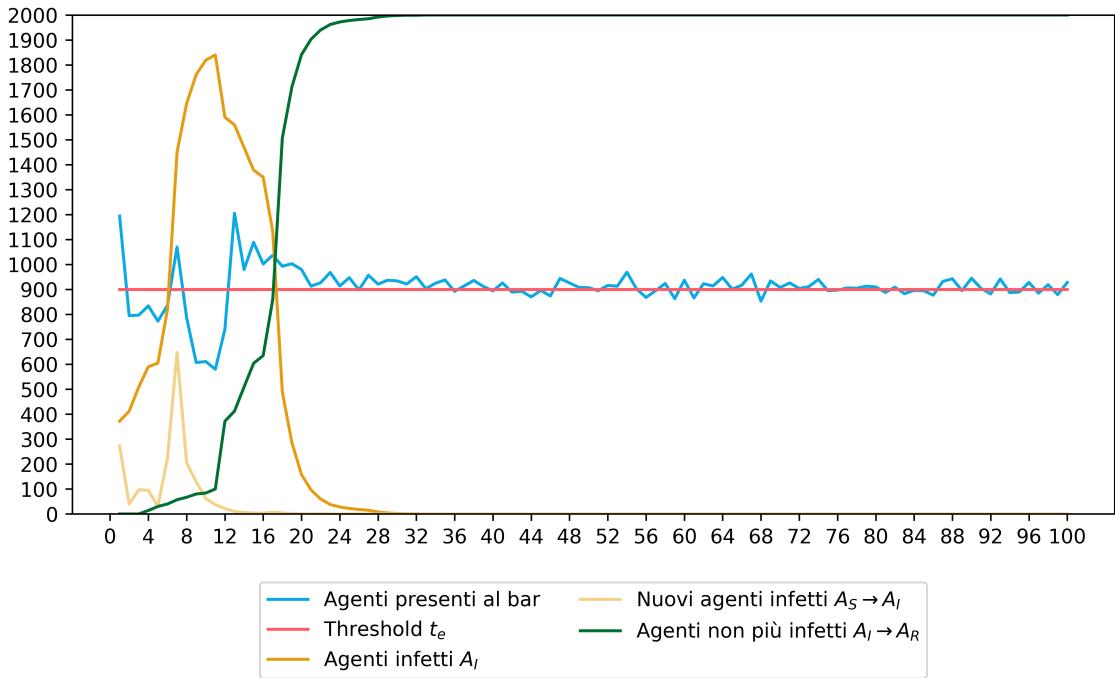


Figura 5: Simulazione epidemia in modalità SIR ( $seed = 80046$ ,  $\alpha = 0, 1$ )

Il modello implementato può rappresentare una dinamica SIR o SIRS (Figura 4) a seconda del parametro  $t_r$  (infection\_cantStartUntil). In un modello SIRS gli agenti recuperati  $A_R$  possono essere infettati nuovamente, tornando dopo un certo periodo di tempo allo stato  $A_S$ , mentre in un modello SIR gli agenti recuperati non possono più essere infettati e rimangono i restanti giorni nello stato  $A_R$ .

La Figura 5 illustra il funzionamento del modello in modalità SIR con  $t_r = 0$ . Come si può osservare dalla curva che rappresenta gli agenti  $A_R$  ( $A_I \rightarrow A_R$ ), il valore iniziale è pari a zero e cresce fino a raggiungere il massimo di duemila. Questo indica che, nel corso della simulazione, tutti gli agenti sono stati contagiati e sono successivamente transitati allo stato di recupero. Nella simulazione mostrata sono stati utilizzati i parametri default, ad eccezione di  $\alpha$  che è stato impostato a 0,1 per meglio mostrare una diffusione più lenta del contagio.

Nelle successive simulazioni il modello verrà utilizzato in modalità SIRS con  $t_r = 3$ .

#### 4.2.1.4 Analisi di sensibilità epidemia

Nel modello sono presenti alcuni parametri che influenzano la diffusione dell'epidemia. In particolare i parametri:

- $\alpha$ : È un parametro amplificatore che rende la diffusione dell'epidemia più rapida.
- $t_r$ : Rappresenta il tempo per cui un agente rimane nello stato  $A_R$ , quindi quel tempo a seguito di un contagio in cui l'agente non può essere reinfettato. Aumentando questo parametro si rende più difficoltosa la diffusione dell'epidemia in quanto diminuiscono gli agenti  $A_S$ .

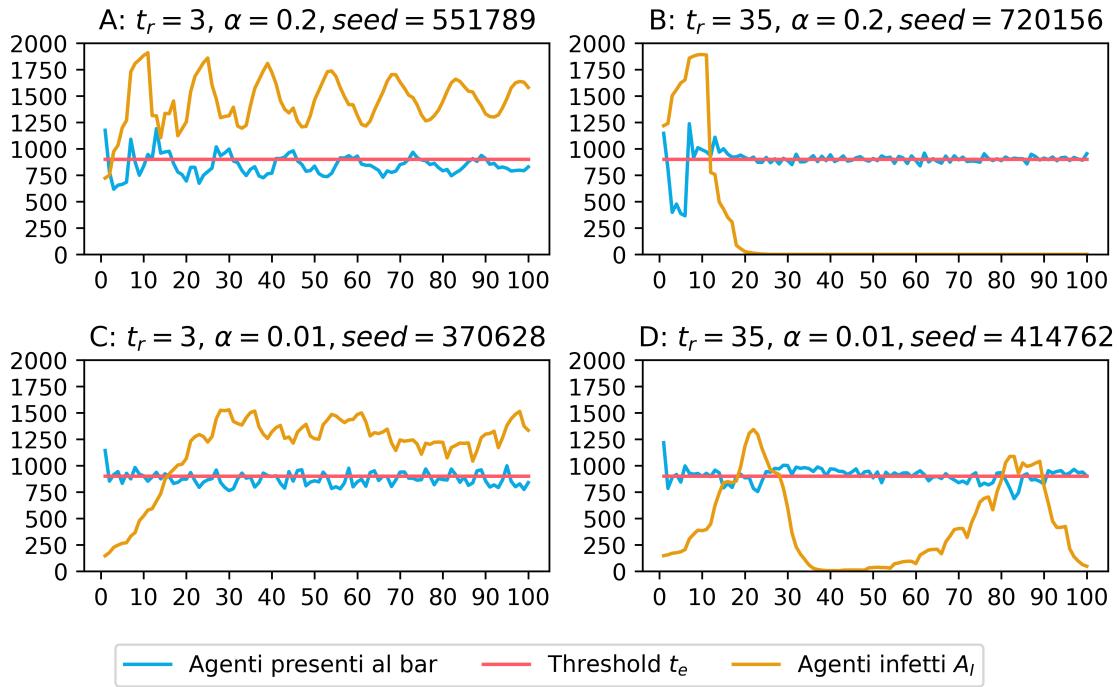


Figura 6: Analisi di sensibilità epidemia

In Figura 6 sono rappresentati i risultati di un'analisi di sensibilità in cui vengono variati i parametri  $\alpha$  e  $t_r$ . Come è possibile notare, aumentando il parametro  $\alpha$  viene aumentata la diffusione del contagio, mentre aumentando il parametro  $t_r$  si ha una diminuzione del numero di agenti infetti.

Nella simulazione  $D$ , si osserva che con l'aumento sia del tempo  $t_r$  e una riduzione del parametro  $\alpha$ , si verifica un primo picco di contagio a causa della rapida diffusione del virus. Successivamente, il contagio entra in una fase dormiente, poiché a causa del precedente picco il numero di agenti nello stato  $A_i$  diminuisce significativamente. Si osserva poi un secondo picco di contagio, dovuto al fatto che gli agenti recuperati  $A_R$  possono essere infettati nuovamente ( $A_R \rightarrow A_S \rightarrow A_I$ ).

In  $C$  si può notare come la diffusione del virus sia particolarmente lenta, la popolazione arriva ad essere in gran parte infetta solo dopo 30 iterazioni  $t$ . Questo è dovuto al fatto che il parametro  $\alpha$  è molto basso, quindi la diffusione del virus è più lenta rispetto agli altri casi. È possibile infatti notare che in  $A$ , essendo  $\alpha$  più alto rispetto  $C$ , l'epidemia si diffonde più rapidamente.

In  $B$  si osserva che la combinazione di un valore elevato di  $\alpha$  e di un alto  $t_r$  interrompe completamente la diffusione del contagio. In questo caso, la combinazione dei due parametri non favorisce la diffusione dell'epidemia. Gli agenti si infettano rapidamente, ed in quanto la durata dello stato  $A_I$  è pari a dieci istanti  $t$  ( $t_i = 10$ ), nessun agente riesce ad essere ancora infetto dopo i 35 istanti  $t$  richiesti da  $t_r$ .

#### 4.2.1.5 Agenti

In riferimento alla Sezione 2.2.2, gli agenti nel modello possiedono diverse caratteristiche; i seguenti punti offrono alcuni esempi rappresentativi per ciascuna di queste caratteristiche:

- **Univoco:** In quanto ogni agente è univoco e possiede attributi e caratteristiche proprie. Il vettore di memoria strategica di un agente  $a$  sarà necessariamente costituito da esiti diversi rispetto a quello di qualsiasi altro agente.
- **Autonomi:** In seguito verrà meglio descritto il processo di decisione relativo alla presenza al bar, la possibile interazione con il sistema bar a seconda di una strategia personale è un esempio di autonomia.
- **Socialità:** La presenza di un agente infetto abilitato al contagio all'interno del bar potrebbe influenzare la probabilità di presenza di altri agenti non ancora contagiati.
- **Adattabilità:** Successivamente verrà illustrato come gli agenti nel modello prendano decisioni basate anche sul proprio vettore di memoria, apprendendo dalle esperienze passate.

L'agente, inoltre, può essere formalmente descritto come un automa, egli infatti può trovarsi in stati diversi in ogni istante  $t$ . L'agente è quindi modellabile attraverso un automa a stati finiti (FSA), il quale può trovarsi in uno dei seguenti stati:

- $S$ : Non presente al bar, suscettibile (agente nello stato  $A_S$ );

- $I$ : Non presente al bar, infetto (agente nello stato  $A_I$ );
- $R$ : Non presente al bar, recuperato (agente nello stato  $A_R$ );
- $S_{bar}$ : Presente al bar, suscettibile (agente nello stato  $A_S$ );
- $I_{bar}$ : Presente al bar, infetto (agente nello stato  $A_I$ );
- $R_{bar}$ : Presente al bar, recuperato (agente nello stato  $A_R$ );

La simulazione può concludersi mentre un agente si trova in uno stato qualunque; per tale ragione, l'automa è composto esclusivamente da stati finali. La simulazione, infatti, si interrompe al raggiungimento del numero massimo di istanti  $t$ , ovvero quando  $t = t_{max}$ , indipendentemente dallo stato degli agenti. L'FSA dell'agente può quindi ricevere in input gli elementi dell'alfabeto  $\Sigma_a = \{b, s\}$ , dove:

- $b$ : L'agente si presenta al bar o esce dal bar;
- $s$ : L'agente passa al prossimo stato SIRS;

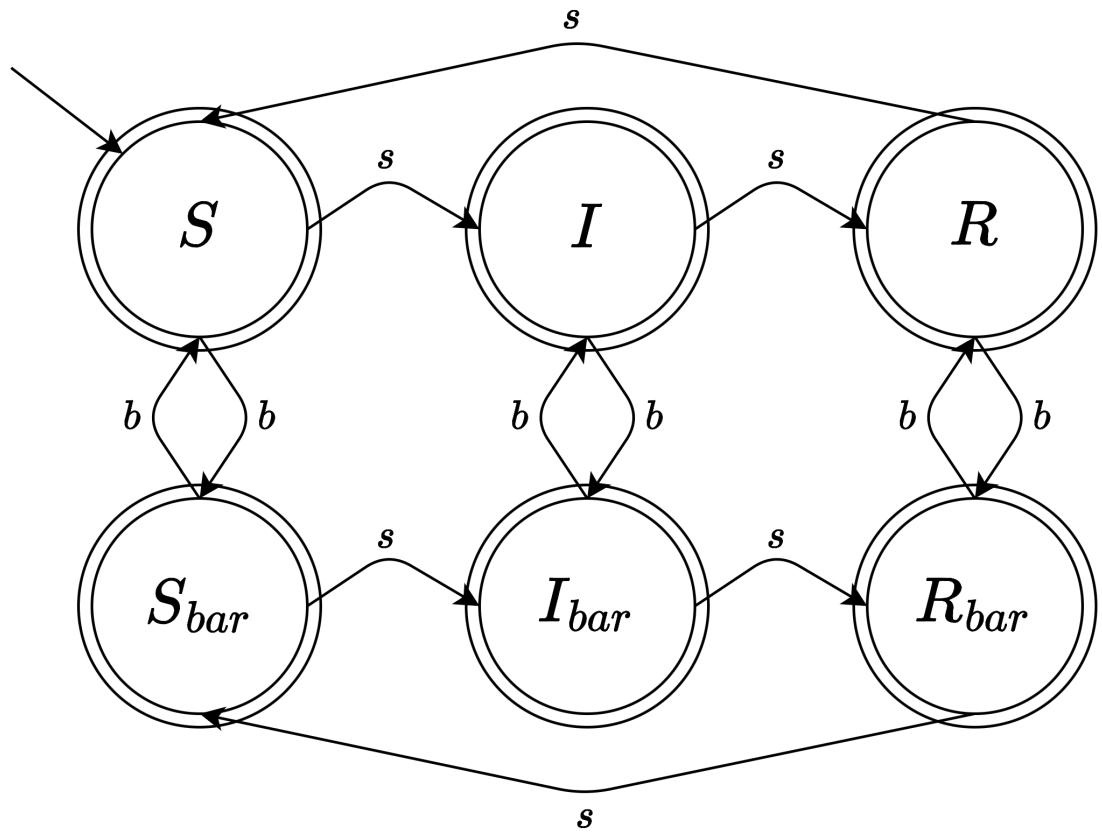


Figura 7: FSA agente  $a$

È stato realizzato il diagramma mostrante l'automa a stati finiti dell'agente in Figura 7. Si formalizza la lingua dell'automata agente come:

$$L = \{w \in \{\varepsilon, b^n, s^n\}^* \mid n \in \mathbb{N}\}$$

#### 4.2.1.6 Policy Maker

Come è possibile notare in Figura 3 ed in Figura 6 [ $A, C$ ], in alcune simulazioni l'epidemia prende il sopravvento infettando gran parte degli agenti ( $\|A_I\| \approx n$ ). In questi scenari l'epidemia potrebbe procedere in modo incontrollato, e gli agenti continuerebbero il ciclo SIRS all'infinito.

Per rendere il modello più realistico, soprattutto in situazioni di epidemie gravi, è stata introdotta l'entità del Policy Maker (PM). Questa figura potrebbe rappresentare il gestore del bar oppure, ampliando il modello a una scala più vasta, addirittura un governo nel caso in cui il bar fosse considerato come uno stato o una regione.

Il PM ha il compito di limitare la diffusione dell'epidemia attraverso l'adozione di misure di contenimento. Le misure implementate nel modello sono le seguenti:

- **$a_1$  (Riduzione capacità):** Il PM può ridurre la capacità massima del bar, in modo da limitare il numero di agenti che si recano al bar, limitando quindi il numero di agenti infetti al bar e il numero di agenti infettabili nel bar.
- **$a_2$  (Dispositivi di protezione delle vie respiratorie):** L'utilizzo di mascherine può limitare la diffusione dell'epidemia. Nel modello sono stati implementati due tipi di mascherine, uno più efficiente dell'altro. Nel modello è implementata la possibilità che un determinato numero di agenti possa introdursi nello spazio senza mascherina aggirando i controlli.
- **$a_3$  (Test ingresso):** Il test di ingresso vieta l'accesso agli agenti con un livello di contagio  $c_a$  superiore ad una certa soglia. Questo permette di limitare l'accesso al bar degli agenti infetti. È possibile che il test in certi casi fallisca, in quanto è stata implementata una percentuale di errore.

In questa prima parte il PM attiva le azioni al raggiungimento di threshold impostati come parametri. Il comportamento del PM sarà oggetto di argomento in Sezione 5 dove verrà discussa l'implementazione di algoritmi di reinforcement learning.

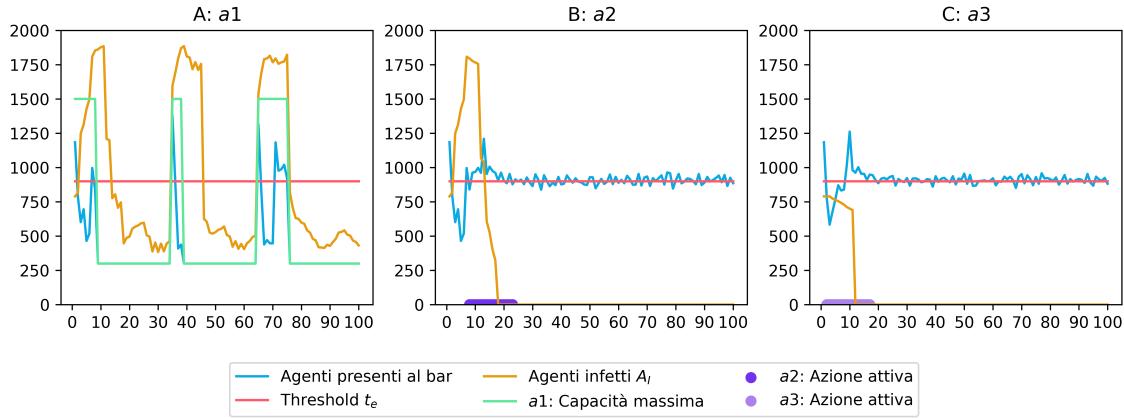


Figura 8: Policy Maker El Farol Epidemic ( $seed = 2002$ )

	A: $a1$	B: $a2$	C: $a3$
Somma del numero di agenti presenti al bar all'istante $t \sum_{t=0}^{100} Att_t$	41.257	90.169	90.881
Somma del numero di agenti infetti all'istante $t \sum_{t=0}^{100} i_t$	91.297	19.955	8.229
Tempo per l'eradicazione del contagio	$\infty$	10	10
Costi totali $\sum_{t=0}^{100} c_{tot_t}$	1.445.254	431.570	203.350

In Figura 8 è possibile osservare il Policy Maker in azione. Utilizzando i parametri legati alle varie azioni impostati nelle simulazioni, in alcuni casi egli riesce a fermare la diffusione dell'epidemia con successo.

In  $A$  limita l'epidemia riducendo la capacità del bar, mentre riesce a fermarla in  $B$  utilizzando i dispositivi di protezione delle vie respiratorie e in  $C$  utilizzando il test di ingresso. Senza considerare i costi associati alle azioni, l'azione  $a3$  appare la più efficace per la risoluzione del contagio, poiché permette un maggior numero di agenti presenti al bar  $Att$  e, allo stesso tempo, il minor numero di agenti contagiati  $i$ . Sia l'azione  $a2$  che l'azione  $a3$  riescono a interrompere il contagio in 10 iterazioni, mentre l'azione  $a1$  non riesce a bloccarne la diffusione, ma solo a limitarne l'intensità. Sebbene le azioni  $a2$  e  $a3$  abbiano applicazioni differenti, le loro capacità di riduzione del contagio sono simili, mentre l'azione  $a1$  differisce maggiormente in tal senso. Si hanno quindi due azioni simili con costi diversi ( $a2$ ,  $a3$ ) e un'azione con costi e risultati differenti ( $a1$ ). La maggior efficacia di  $a3$  rispetto a  $a2$  dipende dal fatto che  $a3$  viene attivata quando la percentuale di contagiati supera il 30% (secondo i parametri di default:  $\frac{i}{n} \geq a3\_InfectedThreshold \mid a3\_InfectedThreshold = 0.3$ ),

mentre  $a2$  si attiva più tardi, al superamento di una soglia di contagio più alta (secondo i parametri di default:  $\frac{i}{n} \geq a2\_InfectedThreshold \mid a2\_InfectedThreshold = 0.85$ ). Questa differenza da tempo al contagio nel caso  $B$  di diffondersi maggiormente, e di conseguenza aumentare il numero di infetti in simulazione.

L'azione  $a3$  risulta essere la più vantaggiosa anche dal punto di vista economico, poiché i costi totali associati alla simulazione  $C$  si dimostrano inferiori rispetto a quelli derivanti dalle altre azioni. In questa sezione, tuttavia, non si effettua un'analisi dettagliata dei costi associati alle singole azioni; tale aspetto verrà approfondito in Sezione 5. Ciò dipende dal fatto che i costi generati dalla simulazione sono fortemente influenzati dalle soglie di attivazione delle azioni (*InfectedThreshold*), pertanto, verranno successivamente esplorati metodi di attivazione delle azioni finalizzati alla minimizzazione dei costi.

Il policy maker è un agente che può utilizzare una o più azioni in ogni istante  $t$ , identificando ogni possibile combinazione di azioni con uno stato univoco, è possibile descrivere il policy maker con un push down automa (PDA) o con un automa a stati finiti (FSA) in caso non si tenga conto del parametro *reductionDuration* delle azioni.

Nel caso di un Final State Automata Ipotizziamo che il pm possa passare tra i seguenti stati:

- $s_{pm_0}$ : Nessuna strategia attiva;
- $s_{pm_1}$ :  $a1$  attiva;
- $s_{pm_2}$ :  $a2$  attiva;
- $s_{pm_3}$ :  $a3$  attiva;
- $s_{pm_4}$ :  $a1$  e  $a2$  attive;
- $s_{pm_5}$ :  $a1$  e  $a3$  attive;
- $s_{pm_6}$ :  $a2$  e  $a3$  attive;
- $s_{pm_7}$ : Tutte le strategie sono attive;

Come per l'automa a stati finiti dell'agente, anche il policy maker è composto esclusivamente da soli stati finali, in quanto la simulazione termina a  $t = t_{max}$  indipendentemente dallo stato del policy maker. In input l'automa riceve gli elementi appartenenti al seguente alfabeto  $\Sigma_{pm} = \{a, b, c, d, e, f\}$ , dove:

- $a$ : Attiva  $a1$ ;
- $b$ : Attiva  $a2$ ;
- $c$ : Attiva  $a3$ ;
- $d$ : Disattiva  $a1$ ;
- $e$ : Disattiva  $a2$ ;
- $f$ : Disattiva  $a3$ ;

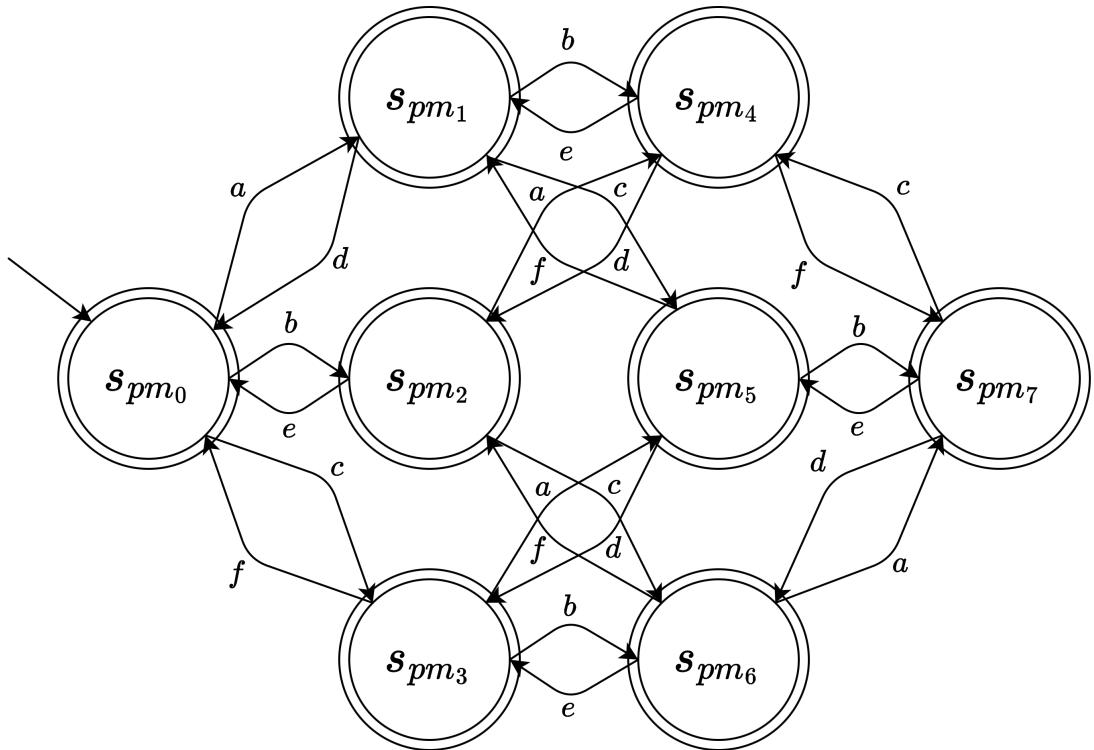


Figura 9: FSA policy maker

Possiamo quindi formalizzare il linguaggio dell'automa visualizzato in Figura 9 come:

$$L = \{w \in \{\varepsilon, a^n, b^n, c^n, d^n, e^n, f^n\}^* \mid n \in \mathbb{N}\}$$

#### 4.2.2 Variabili di stato

##### 4.2.2.1 Variabili dell'agente

L'entità agente è composta dalle seguenti variabili di stato:

##### Parametri El Farol Bar

- $s_a$  (**strategyType**): Indica la strategia che l'agente seguirà in merito alla presenza al bar di El Farol, se pari a 1 allora seguirà la strategia randomica, altrimenti seguirà la strategia basata sulla regressione lineare. Questo parametro viene impostato in fase di generazione.
- $m_a$  (**person\_memory**): È un array contenente le strategie seguite dall'agente nei giorni precedenti e rappresenta la memoria dell'agente.

##### Parametri epidemia

- $c_a$  (**levelContagious**): Rappresenta il livello di contagio dell'agente  $a$ , spazia tra valori di 0 ed 1 e viene aggiornato ogni istante  $t$ .

#### 4.2.3 Variabili globali

##### 4.2.3.1 Variabili del modello

Nel modello sono presenti le seguenti variabili:

- $Att$  (**attendance**): Indica il numero di agenti che si reca al bar in un istante  $t$ .
- $a_{arr}$  (**present\_agents**): È il vettore contenente le entità agente presenti al bar in un istante  $t$ .
- $s_b$  (**present\_agents\_strategy**): Numero che indica il grado di riempimento del bar in un istante  $t$ , viene calcolato come  $s_b = \frac{Att}{c_b}$ .
- $i$  (**n\_infected\_agents**): Numero di agenti infetti in un istante  $t$ .
- $i_{new}$  (**n\_new\_infected**): Rappresenta il numero di agenti contagiati all'istante  $t$ , quindi coloro che in  $t$  vengono contagiati ( $A_S \rightarrow A_I$ ).
- $s$  (**n\_new\_susceptible**): Numero di agenti che passano allo stato suscettibile all'istante  $t$  (nel caso di utilizzo SIRS:  $A_R \rightarrow A_S$ ).
- $r$  (**recovered\_agents**): Numero di agenti recuperati all'istante  $t$ , quindi coloro che in  $t$  smettono di essere contagiati ( $A_I \rightarrow A_R$ ).

##### 4.2.3.2 Parametri modello

Di seguito vengono spiegati tutti i parametri presenti nel modello, la descrizione puntuale dei parametri è situata in appendice (Sezione 7.1). I parametri si suddividono nelle seguenti categorie:

- **Parametri simulazione**: Generici parametri di una simulazione ABM come il numero di istanti e il numero di agenti.
- **Parametri sociali**: Sono i parametri che riguardano il problema del bar El Farol come il threshold di presenza o la capacità massima del bar.

- **Strategie degli agenti:** Parametri che estendono il problema di El Farol Bar differenziando la strategia classica degli agenti introducendo la possibilità di prendere decisioni tramite il vettore memoria.
- **Parametri epidemiologici:** Sezione riguardanti i parametri associati all'epidemia, spaziano dal numero iniziale dei degli agenti infetti al parametro che modula l'intensità dell'epidemia.
- **Parametri del Policy Maker:** In questa sezione vengono descritti i parametri associati al policy maker e nelle successive i parametri associati alle azioni applicabili. Parte di essi vengono considerati solo quando la modalità di reinforcement learning è disattivata.
- **Reinforcement Learning PM:** Parametri associati alla funzionalità di reinforcement learning del policy maker.

#### 4.2.4 Descrizione dei processi e composizione

##### 4.2.4.1 Diffusione dell'epidemia

Di seguito viene spiegata la funzione che gestisce la diffusione dell'epidemia tra gli agenti presenti, verrà in seguito spiegata la funzione che gestisce la presenza al bar. Per il calcolo dei nuovi infetti all'istante  $t$  vengono in primis calcolate i valori associati alle seguenti variabili:

- $C_a$  **contagious\_level\_sum:** Rappresenta la somma dei livelli di contagio  $c_a$  di tutti gli agenti presenti al bar:  $contagious\_level\_sum = \sum_{i=0}^a c_{a_i}$
- $i_b$  **n\_infected\_agents:** Numero di agenti infetti presenti al bar all'istante  $t$ . Questo numero identifica il numero di elementi presenti nel sottoinsieme di agenti allo stato  $A_I$  presenti al bar ( $i_b = \|A_I\| \mid A_I \subset A$ ).
- $s_b$  **n\_susceptible\_agents:** Numero di agenti suscettibili presenti al bar all'istante  $t$ . Questo numero identifica il numero di elementi presenti nel sottoinsieme di agenti allo stato  $A_S$  presenti al bar.

Viene in seguito calcolato il numero di possibili nuovi agenti infetti per agente contagioso:

$$i_{new_a} = \lfloor 0,5 + \left( \frac{\alpha \cdot C_a \cdot s_b}{a} \right) \rfloor$$

Dove:

- $\alpha$ : È un parametro modulatore che rappresenta la forza dell'epidemia
- $Att$ : Rappresenta il numero di agenti presenti al bar (attendance)

---

**Algoritmo 2** Dinamica contagio

---

**Input:** Variabili globali  $gv$ , parametri  $par$

```

for  $i_b$  in  $A_{I_c}$  do
    for  $i = 0$  to  $random.randint(0, i_{new_a})$  do
         $a\_to\_infect \leftarrow random.choice(present\_susceptible\_agents)$ 
        if  $a\_to\_infect.is\_not\_protected$  and  $par.enablePM$  and  $gv.a2\_is\_active$  then
             $present\_susceptible\_agents.remove(a\_to\_infect)$ 
             $infection\_result \leftarrow a\_to\_infect.initiateContagius()$ 
            if  $infection\_result$  then
                 $i_{new} \leftarrow i_{new} + 1$ 
            end if
        end if
    end for
end for

```

---

Un agente nello stato  $A_I$ , indipendentemente dal risultato della propria strategia, non è detto che possa presentarsi al bar. Infatti il proprio livello di contagio  $c_a$  deve essere necessariamente inferiore al threshold  $t_s$  ( $c_a \leq t_s$ ). Agenti per cui la precedente diseguazione non si verifica, non si presentano al bar in quanto i sintomi dell'infezione sono troppo elevati. Inoltre, non tutti gli agenti infetti presenti al bar possono infettarne di altri. Infatti l'agente infetto necessita di avere un livello di contagio  $c_a$  superiore al threshold di infezione  $t_c$ . È presente infine una casistica che si verifica quando la strategia 3 del Policy Maker  $a3$  è attiva e se  $a3\_testFailUnder$  è minore rispetto a  $t_s$  ( $a3\_testFailUnder < t_s$ ), in questo caso la soglia superiore di presenza al bar si abbassa al valore pari a  $a3\_testFailUnder$  e l'agente si presenterà al bar se e solo se  $c_a \leq a3\_testFailUnder$ . Il parametro  $a3\_testFailUnder$  rappresenta la percentuale di errore del test di ingresso, per cui un livello inferiore di contagio rispetto tale soglia non viene identificato come contagioso.

Considerando le precedenti affermazioni, è possibile identificare gli agenti abilitati al contagio, come coloro per cui si verifica:

$$t_c \leq c_a \leq \min(t_s, (a3\_InfectedThreshold \cdot a3\_is\_active))$$

Possiamo quindi definire un ulteriore stato dell'agente  $a$  che identifica gli agenti abilitati al contagio come  $A_{I_c}$ . L'insieme di agenti in tale stato è necessariamente un sotto insieme dell'insieme contenente gli agenti infetti  $A_I$ .

Un ulteriore ostacolo all'infezione degli agenti è rappresentato dalla strategia  $a2$  del PM, essa infatti se abilitata consente agli agenti di indossare una mascherina, che li proteggerà dal contagio. Un agente nello stato  $A_{I_c}$  potrebbe quindi non infettare  $i_{new_a}$  agenti se incontra agenti protetti da mascherina. La dinamica del contagio viene così ultimata con l'infezione dei singoli agenti presenti al bar. Il numero di nuovi infetti all'istante  $t$  sarà quindi  $i_{new} \leq i_{new_a} \cdot i_b$ . Il valore  $i_{new}$  è necessariamente inferiore o uguale al precedente prodotto in quanto  $i_{new_a}$  identifica il numero massimo di contagi che un singolo agente  $a$  nello stato  $A_{I_c}$  può causare.

Nell'Algoritmo 2 è possibile osservare la dinamica del contagio, che si compone di due cicli for, uno riguardante gli agenti  $A_{I_c}$  ed uno per ogni nuovo agente da infettare. Nell'algoritmo viene evidenziato che il numero effettivo di contagi per ciascun agente  $a$  può variare tra un massimo di  $i_{new_a}$  e un minimo di 0. Questo accade perché un agente  $a$ , abilitato al contagio, può infettare al massimo un numero casuale di agenti compreso tra 0 e  $i_{new_a}$ . Tale numero può poi ridursi ulteriormente in seguito alle azioni del policy maker.

#### 4.2.4.2 Strategia dell'agente

La strategia dell'agente  $s_a$  è indicata da un numero decimale definito su un intervallo tra 0 ed 1. Prima di arrivare al valore di  $s_a$  viene calcolato un valore della strategia intermedio  $s_{a_i}$  seguendo due metodi decisionali:

- **Strategia randomica:** L'agente definisce la propria strategia in modo randomico, in questo caso  $s_{a_i}$  è definito come un numero randomico tra 0 ed 1:

$$s_{a_i} = x, x \in \mathbb{R}, x \in [0, 1]$$

- **Strategia basata sulla memoria:** L'agente definisce la propria strategia basandosi sulla propria memoria (contenenti le strategie e i risultati passati). In questo caso  $s_{a_i}$  è definito come il risultato di una regressione lineare del vettore memoria  $m_a$  insieme ad una componente casuale:

$$s_{a_i} = ((1 - 0,5) \cdot x) + (Y_i \cdot 0,5), x \in \mathbb{R}, x \in [0, 1]$$

Dove  $Y_i$  è il valore stimato tramite regressione lineare del vettore memoria.

Il vettore della memoria  $m_a$  è un array che contiene le precedenti strategie seguite dall'agente. Ad ogni iterazione  $t$  viene aggiunto il valore di  $s_a$  al vettore memoria. Se nello stesso istante  $t$  l'agente si presenta al bar, viene rimosso l'ultimo valore del vettore  $m_a$  ed aggiunto un nuovo valore calcolato sulla base del grado di riempimento del bar:

$$\%_{riempimento} = \frac{Att}{c_b}$$

Per il calcolo di  $s_a$  vengono considerati anche i valori precedenti in memoria, questo avviene indipendentemente dal metodo decisionale scelto in quanto i valori in  $m_a$  possono essere aggiornati con il grado di riempimento totale del bar. Il calcolo della strategia  $s_a$  è quindi definito come una media pesata dove gli ultimi  $n$  valori della memoria hanno un peso maggiore secondo il parametro `people_memory_weight_arr`, mentre gli altri un peso minore relativo. Nei parametri di default del modello il vettore `people_memory_weight_arr` si compone di tre valori e i pesi sono rispettivamente:  $p_n = 0,5$   $p_{n-1} = 0,2$   $p_{n-2} = 0,1$ . Il peso relativo  $p_r$  viene calcolato come differenza tra uno e la somma dei valori del vettore `people_memory_weight_arr` divisa per il numero di valori presenti nel vettore memoria  $\|m_a\|$  meno i valori con peso maggiore, ovvero la lunghezza di `people_memory_weight_arr`.

Nel modello con parametri default viene calcolato come:

$$p_r = \frac{1 - (0,5 + 0,2 + 0,1)}{\|m_a\| - 3}$$

Infine il valore di strategia  $s_a$  viene ottenuto attraverso una media ponderata, dando maggior importanza ai valori più recenti della memoria. Viene infatti ipotizzato che essi possano influire maggiormente sulle decisioni dell'agente rispetto a valori meno recenti.

$$s_a = \frac{(m_a[0] \cdot p_r) + (m_a[1] \cdot p_r) + \dots + (m_a[\|m_a\|-2] \cdot p_{\|m_a\|-2}) + (m_a[\|m_a\|-1] \cdot p_{\|m_a\|-1}) + (m_a[\|m_a\|] \cdot p_{\|m_a\|})}{(p_{\|m_a\|} + p_{\|m_a\|-1} + p_{\|m_a\|-2}) + \sum_{i=0}^{\|m_a\|-3} p_{r_i}}$$

#### 4.2.4.3 Presenza al bar

La presenza al bar viene calcolata eseguendo una funzione per ogni singolo agente. Essa tiene conto della strategia e del livello di contagio dell'agente, infatti l'agente si presenta al bar se e solo se le seguenti condizioni sono verificate:

- $s_a < t_e$ : Per cui l'agente decide di presentarsi al bar se la sua strategia è inferiore alla soglia  $t_e$ . Nel modello classico di El Farol descritto da Brian W. Arthur in [17]

l'agente si presenta al bar se la propria strategia  $s_a$  è inferiore alla soglia  $t_e = 0.6$ , quindi se l'agente ritiene che la percentuale di riempimento del bar sia inferiore al 60%.

- $c_a \leq t_s$ : La disequazione si verifica se il livello di contagio dell'agente  $c_a$  è inferiore al threshold  $t_s$  (come descritto in Sezione 4.2.4.1).

Se l'agente decide di presentarsi al bar e le condizioni inerenti al contagio sono favorevoli, viene aggiunto al numero totale di agenti presenti al bar ed al vettore contenente gli agenti presenti ogni giorno. Di seguito è presente uno pseudo codice ed un diagramma di flusso rappresentante questa decisione:

---

**Algoritmo 3** Decisione di partecipazione al bar con contagio

---

**Input:** Variabili globali  $gv$ , parametri  $par$

```

function decisionAttendingBar

    if  $a \geq c_b$  and  $par.respect\_the\_max$  then
        return            $\triangleright$  Se la capacità massima è raggiunta e rispettata, non tentare di
        partecipare

    end if

     $c_a \leftarrow getContagiousLevel()$             $\triangleright$  Livello di contagio dell'agente inizializzato a 0

     $s_a \leftarrow agentCurrentStrategy()$           $\triangleright$  Restituisce la strategia corrente dell'agente

    if  $s_a < t_e$  and  $c_a \leq t_s$  then
         $presence\_bool \leftarrow \text{True}$             $\triangleright$  L'agente decide di partecipare al bar

        if  $par.enablePM$  and  $par.enableA3$  and  $gv.a3\_is\_active$  then
            if  $c_a > par.a3\_testFailUnder$  then
                 $presence\_bool \leftarrow \text{False}$         $\triangleright$  Il Test non ha ammesso l'agente contagioso
            end if

        end if

    end if

end function return:  $presence\_bool$ 

```

---

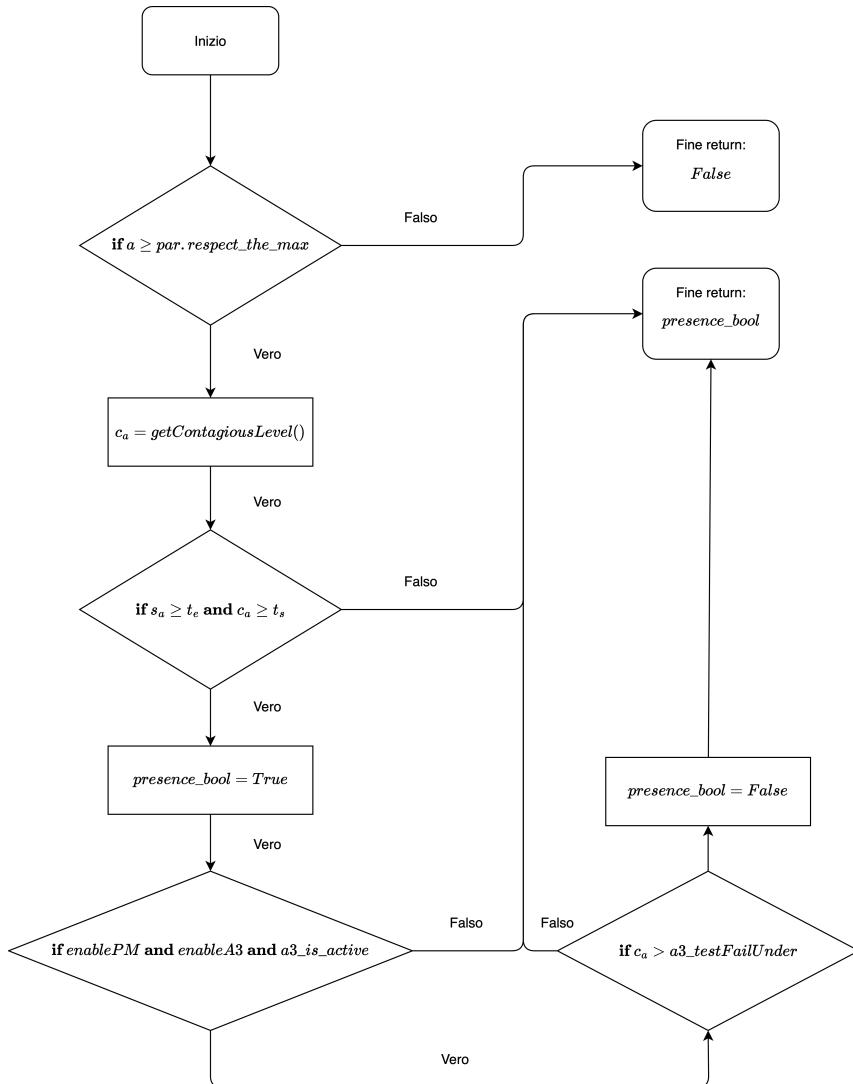


Figura 10: Funzione decisionAttendingBar

#### 4.2.4.4 Gestione delle strategie del Policy Maker

Come descritto in precedenza, il Policy Maker può attuare tre diverse strategie per cercare di ridurre il contagio all'interno del bar.

Nel modello standard, il PM opera applicando le strategie al raggiungimento di threshold definiti come parametro. È quindi possibile che più di una strategia venga attivata contemporaneamente.

Se il grado di agenti infetti su agenti totale supera il threshold di attivazione della strategia allora essa viene attivata. Per l'attivazione di una strategia deve quindi verificarsi:

$$\frac{\|A_I\|}{n} \geq InfectedThreshold$$

I parametri che definiscono l'attivazione delle strategie sono  $a1\_InfectedThreshold$ ,  $a2\_InfectedThreshold$ ,  $a3\_InfectedThreshold$ , rispettivamente per  $a1$ ,  $a2$  e  $a3$ .

Le strategie vengono attivate o disattivate alla fine di ogni istante  $t$  e vengono mantenute attive per un numero di giorni minimo definito dai parametri  $a1\_reductionDuration$ ,  $a2\_reductionDuration$  e  $a3\_reductionDuration$ . Per la disattivazione delle strategie è necessario che sia trascorso il numero di giorni definito dai parametri, sia che la condizione relativa al grado di attivazione sia verificata, altrimenti viene mantenuta attiva per un altro istante  $t$ . Deve quindi verificarsi:

$$t \geq t_f \wedge \frac{\|A_I\|}{n} < InfectedThreshold$$

#### 4.2.4.5 Calcolo del costo degli infetti e azioni del Policy Maker

A diverse variabili del modello sono stati attribuiti dei costi, come descritto nei parametri sono presenti i seguenti costi:

- $c_{a1}$ : Costo dell'azione A1;
- $c_{a3}$ : Costo dell'azione A3;
- $c_{a2_1}$ : Costo dell'azione A2 (tipo 1);
- $c_{a2_2}$ : Costo dell'azione A2 (tipo 2);
- $\delta$ : Costo per ogni nuovo infetto;

Il costo viene calcolato tramite la funzione `calculate_value()` ed ogni istante  $t$  viene aggiunto un valore di costo a quattro vettori, uno per ciascuna voce di costo. Sono presenti anche un vettore di memoria per i valori dei costi totali e costi delle azioni.

I costi vengono calcolati moltiplicando il valore costante di costo unitario con delle variabili:

- $C_{a1}$ : Calcolato come capacità del bar meno capacità attuale, a seguito della riduzione, per il costo unitario dell'azione  $a1$ :  $C_{a1} = (c_b - actual\_capacity) \cdot c_{a1}$
- $C_{a2}$ : Viene calcolato in base al numero di agenti presenti al bar che utilizzano una mascherina di tipo 1 o di tipo 2:  $C_{a2} = ((\|A_{m1_{a2}}\| \cdot c_{a2_1}) + (\|A_{m2_{a2}}\| \cdot c_{a2_2})) \cdot c_{a2_1}$
- $C_{a3}$ : Il costo relativo all'applicazione dell'azione  $a3$  è un costo fisso definito interamente dal parametro  $c_{a3}$ . Nella stessa simulazione il costo viene applicato una volta sola, ipotizzando che il macchinario per i test di ingresso venga acquistato in un solo istante  $t$  e possa essere utilizzato nei restanti.
- $\Delta$ : Viene moltiplicato il numero di nuovi infetti all'istante  $t$  per il costo dei nuovi infetti:  

$$\Delta = i_{new} \cdot \delta.$$

#### 4.2.4.6 Ciclo del modello e flusso generale

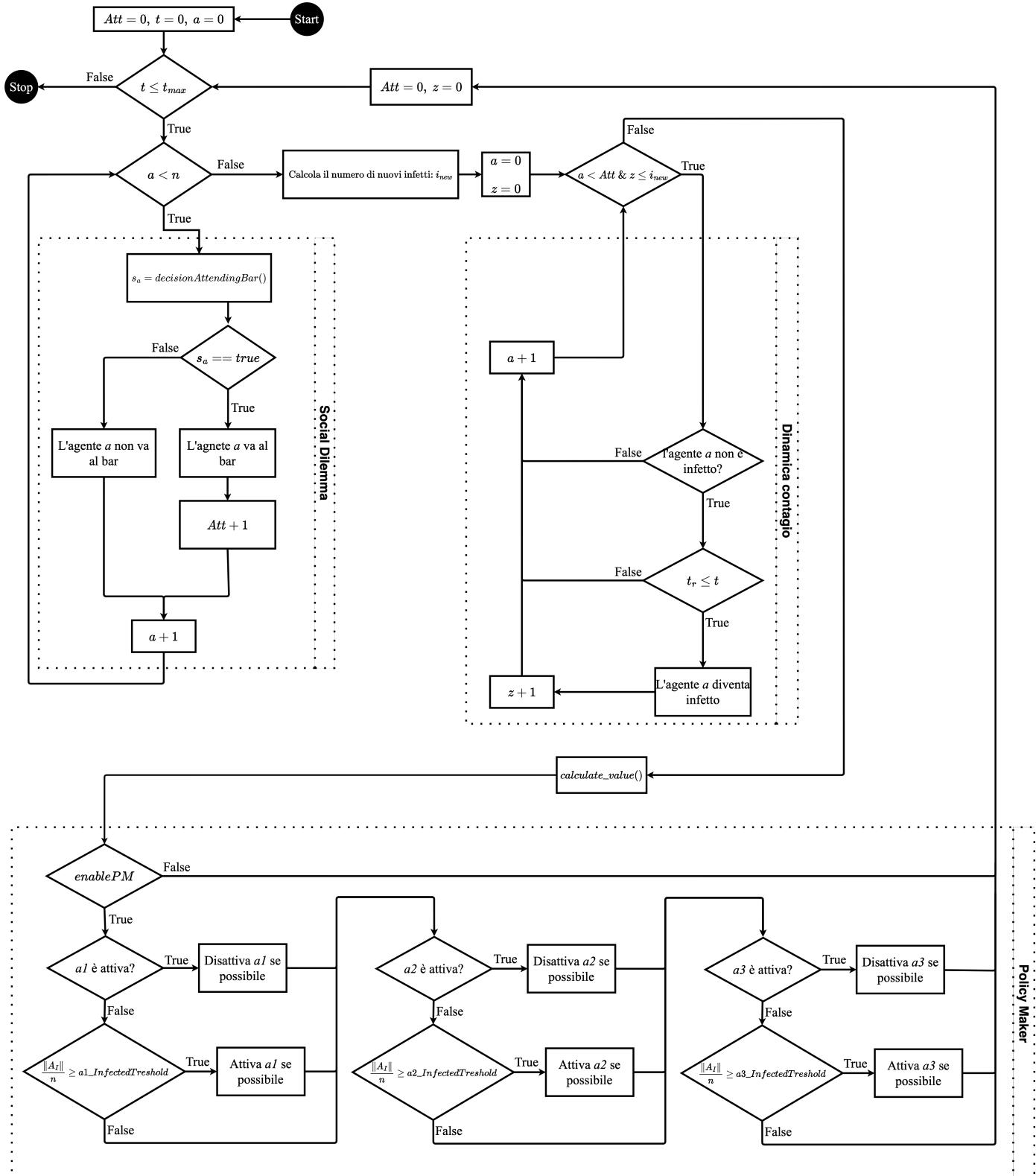


Figura 11: Diagramma di flusso del modello

Ogni istante  $t$  il modello esegue le operazioni ed i processi sopra descritti. La gestione del singolo istante è attribuita alla funzione  $go()$  il cui funzionamento è osservabile in Figura 11, dove viene descritto tramite un diagramma di flusso UML l'ordine di esecuzione dei processi.

### 4.3 Concetti di design

#### 4.3.1 Decoro del livello di contagio

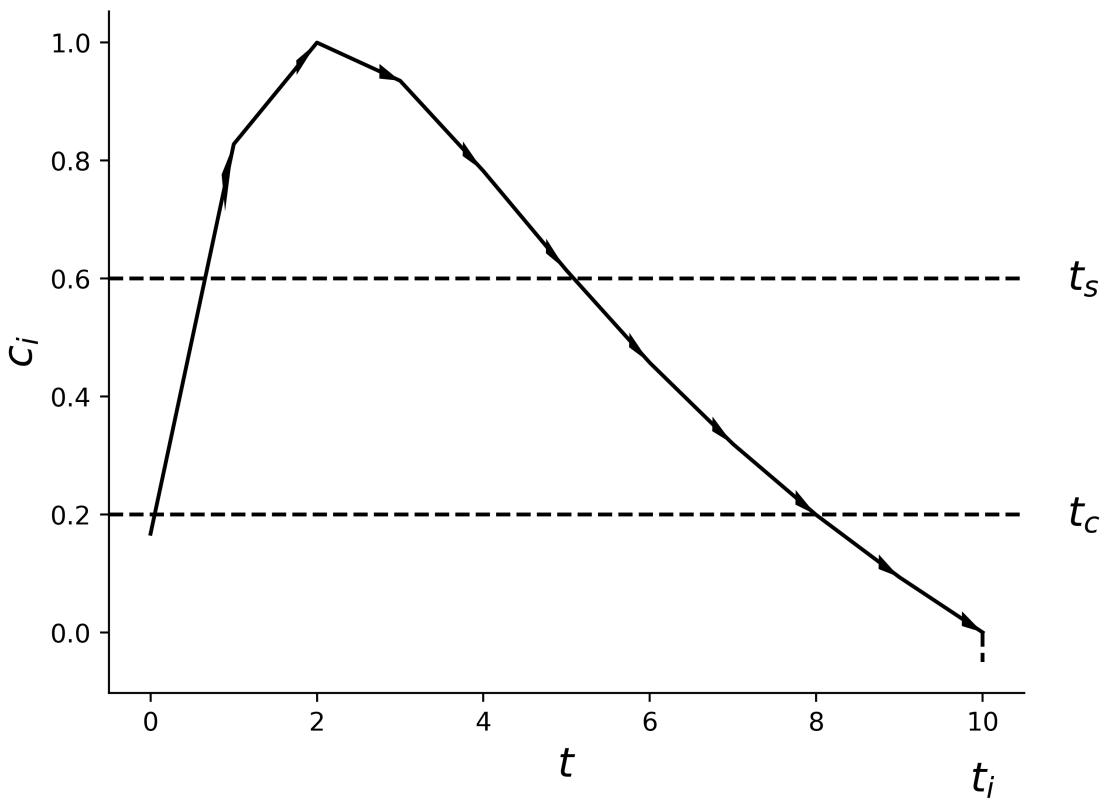


Figura 12: Decoro del livello di contagio

Il livello di contagio  $c_a$  di un agente è un valore che varia tra 0 e 1, e rappresenta la probabilità che l'agente possa infettare un altro agente. Il livello di contagio di un agente varia ad ogni iterazione  $t$  e viene calcolato in base alla funzione:

$$f(x) = 0,6 \cdot \frac{(1-t)^2}{t_i}$$

Essa viene poi modulata per far sì che il suo valore massimo sia pari ad 1, ed il valore minimo pari a 0. Nella configurazione di default il parametro indicante la durata dell'in-

fezione  $t_i$ , ovvero il tempo per cui un agente contagiato permane nello stato  $A_i$  è pari a 10 giorni. È possibile visualizzare il decorso di  $c_a$  su 10 giorni in Figura 12. Inizialmente per identificare il livello di contagio di un agente  $c_a$  veniva utilizzata una retta che partiva da uno e scendeva linearmente a 0. Questo metodo è stato successivamente modificato per rendere il decorso del contagio più realistico, così che su una durata del contagio pari a dieci istanti  $t_i = 10$  il picco massimo del contagio, coincidente con i massimi sintomi e la massima capacità infettiva, venga raggiunto all’istante due.

La funzione `getContagiousLevel()` restituisce il valore di  $c_a$  all’istante  $t$  relativo alla data di inizio contagio.

#### 4.3.2 Stocasticità

Nel modello è presente un elevato grado di stocasticità. L’intera dinamica di El Farol basa le proprie fondamenta su strategie randomiche, almeno all’inizio, degli agenti. Nel modello proposto attraverso l’implementazione della memoria dell’agente, modificabile se si presenta al bar, viene in parte ridotto il grado di stocasticità rispetto al modello classico proposto da Arthur [17].

Nella dinamica di contagio non viene simulata la posizione dell’agente, le interazioni non seguono quindi criteri spaziali o relazionali. Un agente  $a$  appartenente allo stato  $A_I$  che si presenta al bar nell’istante  $t$ , potrebbe collocarsi in una posizione piuttosto che un’altra. Anche nel caso in cui l’agente  $a$  preferisca posizionarsi sempre nel quadrante in alto a sinistra del bar, non è detto che altri agenti suscettibili o infetti abbiano la stessa preferenza. Le interazioni tra gli agenti presenti al bar sono quindi per questo motivo randomiche, un agente  $A_I$  può quindi infettare al massimo  $i_{newa}$  scelti casualmente tra gli agenti presenti, non seguendo quindi criteri spaziali.

È inoltre presente un elemento stocastico nella funzione che restituisce il livello di contagio  $c_a$ , in modo da introdurre del rumore tra i livelli di contagio degli agenti.

Sebbene sia presente un notevole grado di stocasticità, i risultati sono replicabili in quanto viene salvato il seme corrispondente alla simulazione.

#### 4.3.3 Ulteriori fenomeni emergenti

Altri fenomeni emergenti verranno discussi in Sezione 4.5 dove sono situati i risultati ottenuti dal modello. Ulteriori analisi possono essere condotte sul funzionamento di una versione preliminare del modello descritto in [19]. In tale versione, non sono stati imple-

mentati metodi per la riduzione del contagio, e risulta quindi assente la figura del policy maker. Sono state tuttavia effettuate analisi approfondite sui fenomeni emergenti dal modello ad agenti epidemiologico in relazione al problema di El Farol Bar.

## 4.4 Dettagli

### 4.4.1 Inizializzazione

In fase di setup, l'omonima funzione gestisce l'inizializzazione del modello, generando gli agenti necessari per la simulazione. L'inizializzazione degli agenti genera entità che possono differire tra di loro in quanto a:

- **Strategia:** La strategia dell'agente può essere randomica o basata sulla memoria. In base ai parametri del modello viene scelta quale tipologia di strategie seguirà l'agente;
- **Resistenza al contagio:** Sono presenti tre tipologie di agenti, la seconda nel momento del confronto del livello di contagio dell'agente  $S_I$  con il threshold di contagio  $t_c$  lasciano quest'ultimo invariato. Negli altri casi il confronto avviene a seguito dell'incremento o diminuzione del threshold  $t_c$  rispettivamente per la tipologia 1 e 3;
- **Tipologia  $a_2$ :** L'azione  $a_2$  del Policy Maker prevede l'utilizzo di svariate tipologie di mascherine; in questa fase viene definita la tipologia di dispositivo di protezione che verrà assegnata all'agente nel caso in cui l'azione venga abilitata.
- **Infettività:** Ad inizio simulazione viene impostato un numero iniziale di agenti infetti, non è detto quindi che l'agente all'istante  $t = 0$  appartenga allo stato  $A_S$ .

## 4.5 Risultati

### 4.5.1 Introduzione

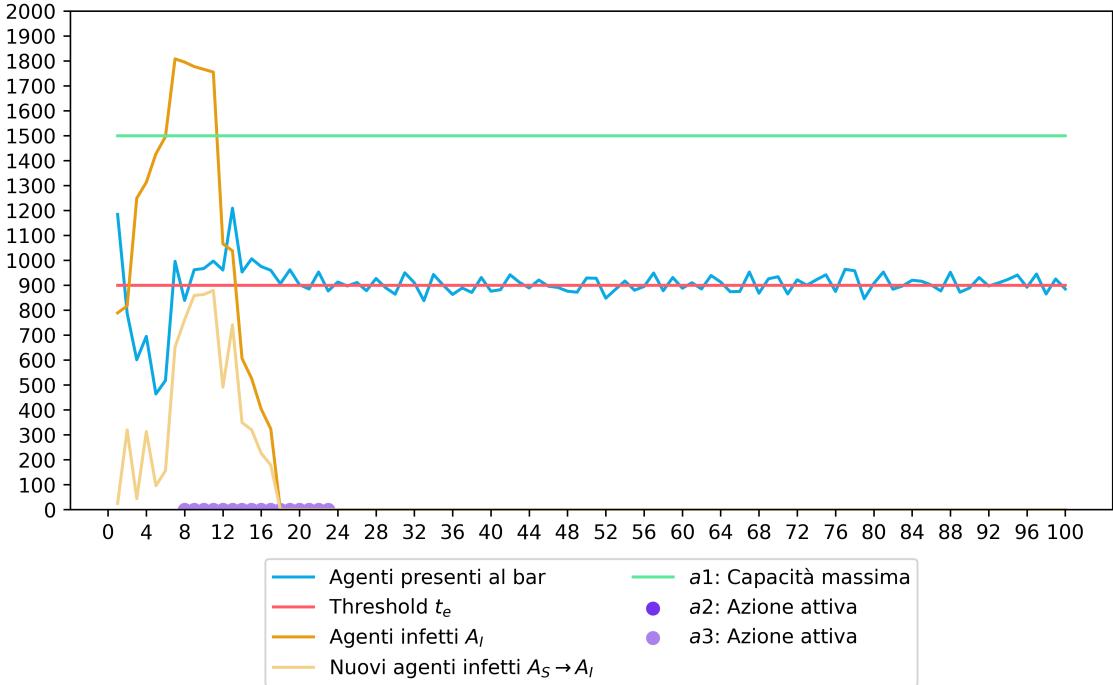


Figura 13: Simulazione modello ( $seed = 2002$ )

In questa sezione vengono analizzati i risultati ottenuti dal modello descritto finora. La Figura 13 riporta i risultati generati da una simulazione che utilizza i parametri default, descritti in Sezione 7.1. Nella simulazione rappresentata, è attivo il policy maker che interviene prontamente per interrompere il contagio attraverso l'applicazione dell'azione  $a_3$ . L'azione  $a_3$  è selezionata poiché è la prima a essere attivata, dato che richiede una soglia di infezione più bassa rispetto ad altre azioni. L'attivazione di  $a_3$  impedisce inoltre che si raggiungano i livelli di contagio necessari per attivare altre azioni, come  $a_1$  o  $a_2$ . Come descritto in Sezione 4.2.1.6, il Policy Maker nel modello descritto fin ora è un elemento passivo, che si limita ad applicare le azioni al raggiungimento di determinate soglie (threshold). Successivamente, verrà illustrata in dettaglio l'applicazione delle azioni basata su un processo decisionale ottenuto tramite l'impiego di tecniche di apprendimento per rinforzo.

## 4.5.2 Fenomeni emergenti

### 4.5.2.1 Effetti del contagio sulla presenza al bar

Nelle analisi riportate in questa sezione, il Policy Maker è stato disattivato al fine di osservare e valutare gli effetti della malattia sulla dinamica del problema di El Farol Bar in assenza di interventi.

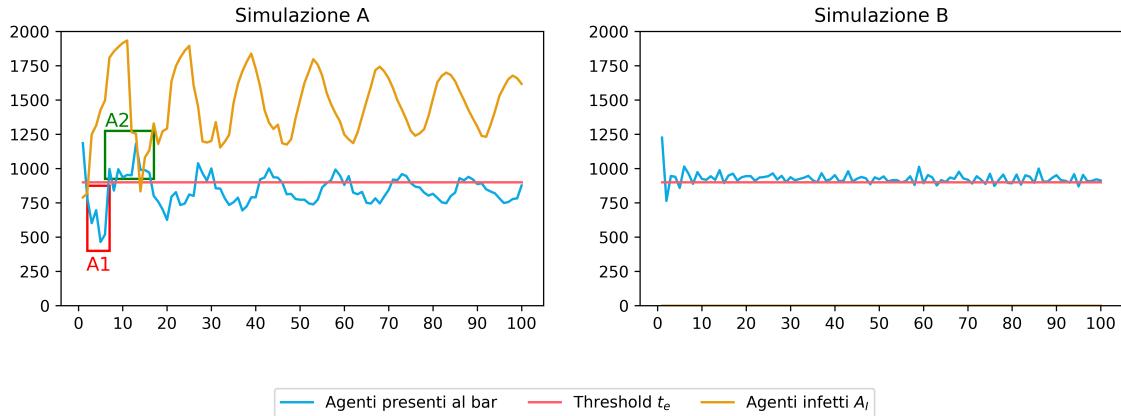


Figura 14: Impatto della malattia sulla presenza al bar ( $seed = 2002$ )

Vengono proposti due scenari visualizzabili in figura Figura 14. In simulazione *A* è abilitato il contagio mentre in simulazione *B* viene temporaneamente disabilitato. A differenza di *B*, in *A* sono state evidenziate due regioni in cui emergono comportamenti a livello di sistema assenti in *B*:

- *A1*: Nella regione evidenziata in rosso si osserva una diminuzione del numero di agenti presenti al bar ( $Att$ ) causata dall'aumento dei contagi. Questo incremento nei contagi porta infatti i livelli di infezione degli agenti ( $c_a$ ) a superare la soglia massima consentita per la presenza al bar ( $t_s$ ), impedendo così l'accesso agli agenti ( $c_a > t_s$ ).
- *A2*: La regione in verde indica un notevole aumento nel numero di agenti presenti al bar ( $Att$ ), ciò è da attribuirsi all'implementazione della memoria agli agenti nel modello. Come descritto in Sezione 4.2.4.2, la strategia adottata dagli agenti ( $s_a$ ) è determinata da una media ponderata considerando i tre elementi più recenti presenti in memoria e i dati restanti. Inoltre, per gli agenti che seguono strategie di secondo tipo, la strategia risultante include un ulteriore componente relativa alle strategie e alle esperienze passate contenute in memoria. In quanto l'ente aggiorna la strategia salvata in memoria con il grado di riempimento del bar se decide di recarsi al bar

all'istante  $t$ , si osserva un aumento del numero complessivo di agenti presenti ( $Att$ ). Questo avviene poiché, a seguito di una minore affluenza al bar, come indicato nel comportamento  $A1$ , i valori più recenti in memoria riflettono una condizione di bar sottoutilizzato (quindi valori di  $s_a \leq t_e$ ), favorendo in seguito la frequentazione del bar da parte degli agenti.

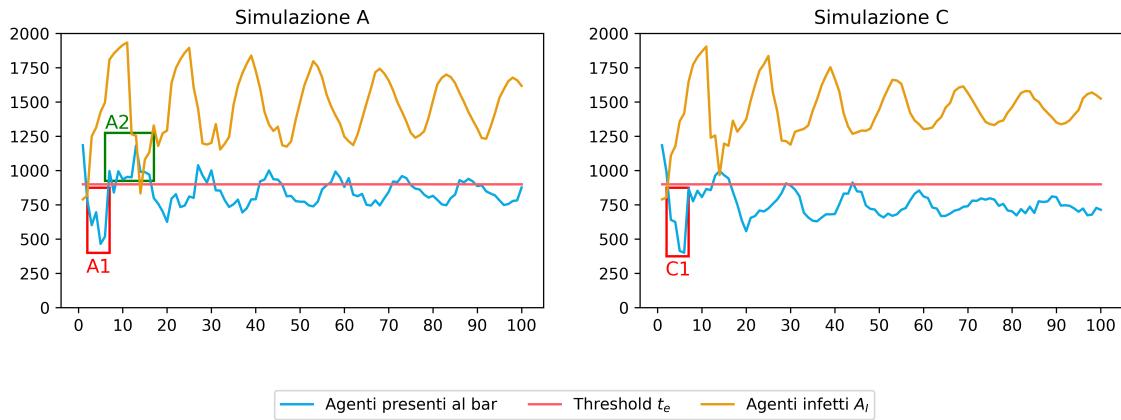


Figura 15: Confronto aggiornamento della memoria dell'agente ( $seed = 2002$ )

Risultati	A	B	C
Media attendance $E(a)$	840,98	927,51	752,32
Media attendance % $\frac{E(a)}{n}$	42,05 %	46,38 %	37,62 %
Media infetti $E(i)$	1459,39	0	1435,96
Media infetti % $\frac{E(i)}{n}$	72,97 %	0 %	71,80 %

Tabella 2: Risultati simulazioni  $A$ ,  $B$  e  $C$  ( $seed = 2002$ )

In Figura 15 viene mostrata una comparazione tra la simulazione  $A$  e una nuova simulazione ( $C$ ), nella quale sono stati disabilitati i meccanismi che portano ad un aumento di  $Att$  in  $A2$ . La disabilitazione è stata implementata impostando manualmente, a ogni istante  $t$ , un grado di riempimento del bar pari alla media delle percentuali di riempimento osservate nella simulazione  $B$ . Così facendo, l'effetto del contagio non incide sulla memoria degli agenti, e quindi non influenza la loro frequenza al bar. Si osserva che, nella simulazione  $C$ , sono presenti solo fasi di riduzione dell'affluenza al bar, una di esse viene indicata come  $C1$ , ma di aumento come in  $A2$ . In altre parole, in  $C$ , l'affluenza al bar può essere modificata solo negativamente, come nel caso di  $A1$ , ma non positivamente,

come avviene in *A2*.

Dai risultati ottenuti (Tabella 2) si osserva che l'utilizzo della memoria degli agenti (simulazione *A*) porta a un aumento della presenza al bar (*Att*) ed a un conseguente, seppur moderato, incremento del numero di infetti, poiché una maggiore affluenza facilita la diffusione del contagio. La simulazione *B*, tuttavia, continua a mostrare il maggior valore di *Att*, non subendo gli effetti negativi legati al contagio descritti nelle condizioni *A1* e *C1*.

L'implementazione della memoria degli agenti è quindi da considerarsi la causa principale per il fenomeno emergente osservato, tale risultato non sarebbe stato ottenibile senza i processi descritti in Sezione 4.2.4.2. Tale fenomeno costituisce un ulteriore elemento distintivo rispetto al modello classico di El Farol proposto da Arthur in [17].

#### 4.5.2.2 Impatto del policy maker sulla presenza al bar

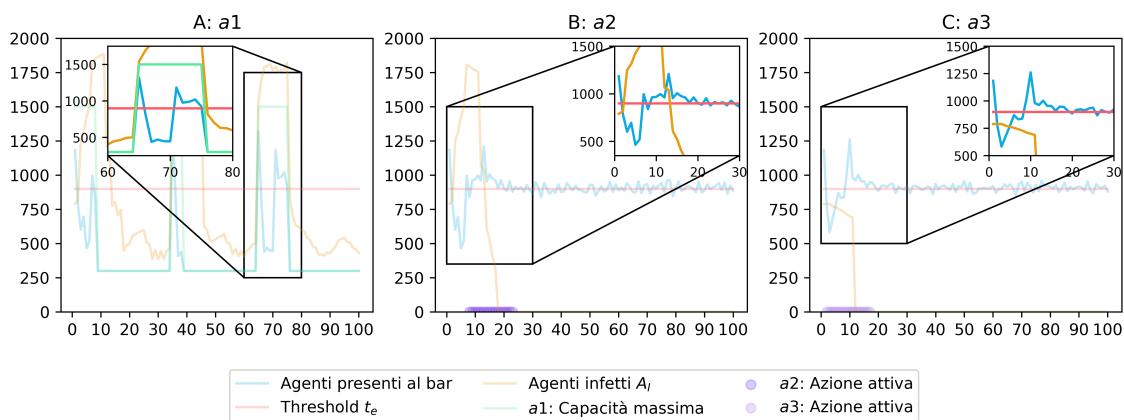


Figura 16: Zoom simulazioni policy maker

La Figura 16 approfondisce i risultati già riportati in Figura 8 e descritti in Sezione 4.2.1.6. Dai grafici, si osserva come il policy maker influenzi la presenza degli agenti al bar (*Att*) in modo simile a quanto descritto nella precedente Sezione 4.5.2.1. In particolare, si nota che:

- *a1*: Riduce la capacità massima del bar, portando ad una diminuzione del numero di agenti presenti al bar *Att*;
- *a2*: Non produce effetti sul numero di agenti presenti al bar *Att*;
- *a3*: Può ridurre il numero di agenti infetti al bar *Att* quando un agente sufficiente infetto viene rilevato dal test di ingresso.

Sebbene le strategie  $a_2$  e  $a_3$  non generino effetti aggiuntivi oltre a quelli legati al contagio, l'attivazione della strategia  $a_1$  produce fenomeni distinti. Nella simulazione  $A$ , l'azione  $a_1$  riduce temporaneamente la capacità massima; una volta conclusa e ripristinata la capacità ordinaria, si verifica un aumento del numero di agenti presenti al bar  $Att$ . Questo fenomeno è analogo a quanto descritto nella Sezione 4.5.2.1 per la regione  $A2$  ed è strettamente connesso all'implementazione della memoria degli agenti. L'aumento improvviso di  $Att$  comporta un incremento dei contagi, poiché sia gli agenti infetti sia quelli non infetti sono incentivati per effetto della memoria a recarsi al bar. Subito dopo questo picco di  $Att$ , si osserva quindi un aumento nel numero di agenti infetti  $i$  e una conseguente successiva riduzione di  $Att$ , poiché gli agenti appena contagiati possiedono valori di  $c_a$  superiori al threshold di massimo livello di contagio per la presenza al bar  $t_s$  ( $c_a > t_s$ ). Successivamente, quando per gli agenti si verifica  $c_a \leq t_s$ , il valore di  $Att$  torna alla normalità e il contagio continua a diffondersi.

#### 4.5.3 Confronto con altri modelli

Nel corso del tempo sono stati realizzati diversi modelli ABM che implementano la dinamica di El Farol.

Durante la realizzazione di questa tesi, ho avuto la possibilità di confrontarmi con un mio collega di studi con la quale ho potuto condividere l'interesse per la materia. Anch'egli ha realizzato un modello ABM basato sulla dinamica di El Farol con un contagio, in ateneo abbiamo avuto la possibilità di confrontarci riguardo ai due modelli. In questa sezione si farà riferimento al modello descritto in precedenza come "modello 1" mentre il modello sviluppato dal collega Luca Pasquino verrà identificato come "modello 2".

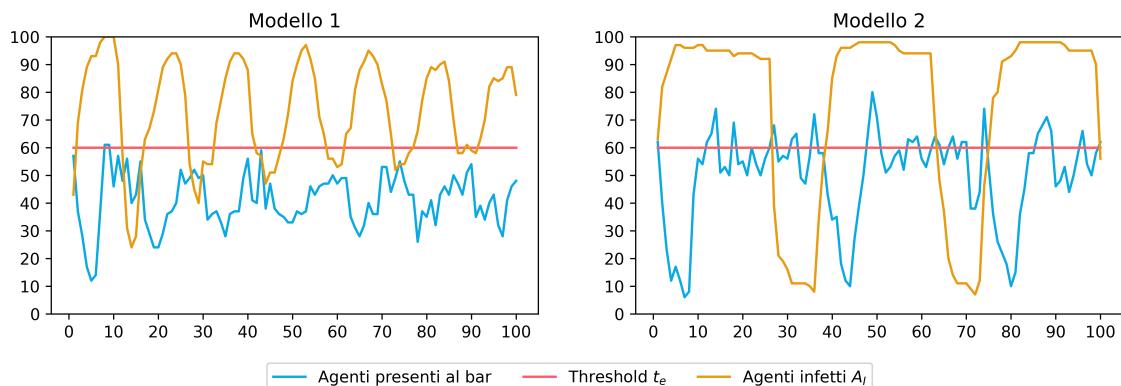


Figura 17: Comparazione modello 1 e modello 2

Come mostrato in Figura 17 inizialmente i due modelli presentavano evidenti differenze. Tramite l'ausilio della descrizione ODD e la condivisione di idee e concetti, siamo riusciti a rendere i due modelli più simili.

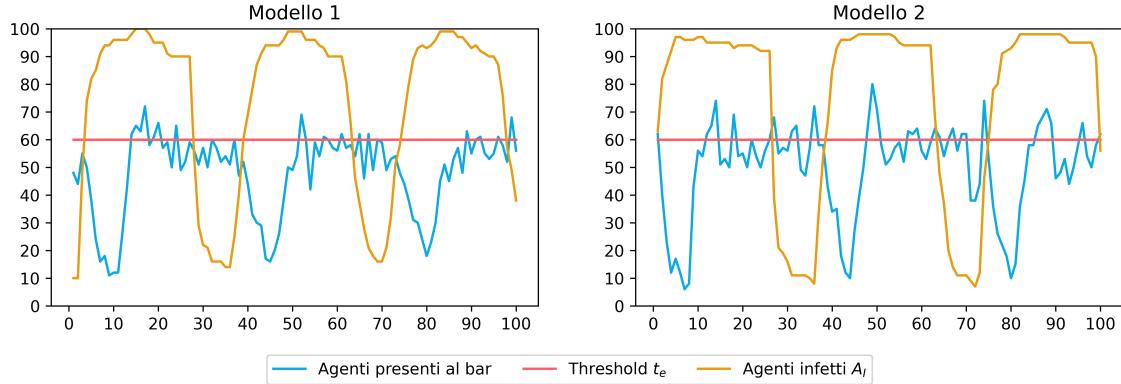


Figura 18: Comparazione modelli con modifica parametri modello 2

In Figura 18 è possibile osservare i risultati ottenuti dai due modelli. Il modello 1 è stato modificato per renderlo più simile al modello 2, in particolare sono stati modificati i parametri:

- $c_b = 75 \rightarrow 100$ : È stato aumentata la capacità massima del bar in quanto analizzando la sezione del metodo ODD riguardo alle variabili di stato si è notato che il modello 2 non presentasse una capacità massima del bar  $c_b$ , è stata quindi posta uguale al numero di agenti nella simulazione  $n$  nel modello 1 in modo da disabilitare tale funzionalità;
- $t_i = 10 \rightarrow 24$ : Aumentato il tempo di infezione;
- $t_r = 3 \rightarrow 9$ : Aumentato il tempo di recupero;
- $\alpha = 0,2 \rightarrow 0,6$ : Rafforzata l'epidemia aumentando il parametro  $\alpha$  che modula la diffusione della stessa.

I parametri sono stati modificati a causa di una diversa interpretazione dei concetti di base del modello, in particolare per quanto riguarda la capacità massima del bar e le equazioni alla base della diffusione del contagio. Alcune diseguaglianze presentavano operatori di confronto “maggiore o uguale” anziché “maggiore”. Per questo motivo, i parametri  $t_i$  e  $t_r$  non sono stati aumentati ad un valore di 25 e 10 (come nel modello 2) ma ad un valore

di 24 e 9. Di seguito è possibile osservare alcuni dei risultati ottenuti dalle simulazioni effettuate per il confronto dei modelli:

Risultati	Modello 1	Modello 2	Modello 1 (modifica parametri)
Media attendance % $\frac{E(Att)}{n}$	40,98 %	49,85 %	48,14 %
Media infetti % $\frac{E(i)}{n}$	73,12 %	75,92 %	73,91 %

La rimozione della capacità massima del bar ha portato ad un aumento della media percentuale di agenti presenti al bar. L'aumento di  $t_r$  e  $t_i$  ha esteso la lunghezza d'onda della curva dei contagi, senza modificare significativamente il valore medio percentuale degli agenti infetti. Questo fenomeno è prevedibile, poiché, in assenza di interventi da parte di un policy maker e in presenza di una malattia sufficientemente contagiosa, la curva dei contagi tende a manifestare un comportamento quasi periodico specialmente nei primi periodi. Se una funzione è periodica e integra la stessa quantità in ogni periodo, l'integrale della funzione su un periodo è invariato rispetto alla durata di quel periodo. Questo è dovuto al fatto che la somma (o l'integrale) di valori ripetuti in una funzione su ogni periodo rimane proporzionalmente costante.

## 5 Reinforcement Learning in ABM

### 5.1 Introduzione

In questa sezione viene descritta l'implementazione della capacità di apprendimento del policy maker, in materia di applicazione delle strategie, per l'ottimizzazione dei costi della simulazione.

Per l'implementazione di tale capacità sono stati utilizzati gli algoritmi di reinforcement learning ampiamente discussi in Sezione 3. I contenuti che seguiranno faranno riferimento al modello epidemiologico descritto nella Sezione 4.

Attivando la modalità di reinforcement learning il policy maker non applicherà più le strategie in base a threshold definiti, ma si affiderà ad una q-table che conterrà i risultati delle azioni applicate in relazione allo stato del sistema.

### 5.2 Panoramica Q-Table

La q-table è lo strumento che rappresenta la memoria del policy maker, essa consente infatti di mappare i risultati delle azioni intraprese dal pm in relazione allo stato. Secondo quanto descritto nella Sezione 3, i risultati  $r$  delle azioni sono identificate da un output generato a seguito dell'applicazione delle stesse, e l'agente  $a$  è rappresentato dal policy maker.

Di seguito verrà fatto riferimento al policy maker stesso con la simbologia  $pm$  quando ci si riferirà all'agente  $a$  i cui risultati delle azioni sono contenute nella q-table.

#### 5.2.1 Composizione

Possiamo quindi definire la composizione della q-table delle azioni applicabili dal  $pm$  ( $Q_{pm}$ ) come una matrice di dimensioni  $S \cdot A$  dove:

- $S$ : Set di stati in cui può trovarsi la simulazione. Gli stati sono definiti dividendo in quartili il numero di agenti che possono essere infetti  $i$  in un'istante  $t$  della simulazione. Per ciascuno dei quartili del livello di contagio, sono stati definiti tre sottostati che dipendono dall'andamento del contagio osservato negli ultimi cinque (numero definito dal parametro  $infection\_slope\_regr\_len = 5$ ) istanti temporali. La classificazione nei sottostati è determinata dalla pendenza ( $\beta$ ) di una retta di regressione

lineare, calcolata sui valori di contagio relativi a tali istanti. In particolare per il primo quartile avremmo:

- $s_{1_1}$ : Stato in cui il contagio è in diminuzione, la pendenza della retta di regressione è negativa  $\beta < 0$ ;
- $s_{1_2}$ : Stato in cui il contagio è stabile, la pendenza della retta di regressione è nulla  $\beta = 0$ ;
- $s_{1_3}$ : Stato in cui il contagio è in aumento, la pendenza della retta di regressione è positiva  $\beta > 0$ ;

Utilizzando lo stesso criterio si possono definire i sotto stati in funzione degli altri quartili del livello di contagio. Il numero di stati è quindi definito come  $\|S\| = 4 \cdot 3 = 12$ , numerando gli stati da uno a dodici seguendo la logica precedentemente esplicitata per il primo quartile:  $s_1 = s_{1_1}, s_2 = s_{1_2}, s_3 = s_{1_3}, s_4 = s_{2_1}, s_5 = s_{2_2}, \dots$ . Definiamo quindi  $S$  come un insieme di stati composto da:  $S = \{s_1, s_2, s_3, \dots, s_{12}\}$ .

- $A$ : Set di azioni che il policy maker può applicare all'istante  $t$  ed in un qualiasi stato del sistema  $s$ . Le tre azioni applicabili dal policy maker sono state precedentemente descritte nella Sezione 4.2.1.6. Definiamo quindi  $A$  un set di lunghezza  $\|A\| = 3$  e definito come  $A = \{a_1, a_2, a_3\}$ .

Una prima definizione della q-table  $Q_{pm}$  identifica quindi una matrice di dimensione  $12 \cdot 3$  con trentasei combinazioni di coppie stato-azione possibili.

### 5.2.2 Aggiornamento Q-Table

L'aggiornamento della matrice delle coppie stato-azione viene effettuata con una versione modificata della "Bellman Optimality Equation":

$$Q_{pm\_new}(s_t, a_t) = (Q_{pm}(s_t, a_t) \cdot (1 - \alpha)) + (r_t \cdot \alpha)$$

Dove  $r_t$  viene ottenuto come somma dei costi totali sostenuti negli istanti in cui l'azione  $a_{pm}$  era attiva. Ad ogni istante  $t$ , tramite la funzione `calculate_value()` precedentemente descritta nella Sezione 4.2.4.5, vengono inseriti in vettori di memoria i valori di costo. Il costo dell'azione  $a$  è calcolato come la somma dei valori di costo presenti nel vettore di memoria dei costi totali, sommando esclusivamente i valori corrispondenti agli istanti in cui l'azione  $a$  è stata attiva. Ciò avviene in quanto il vettore dei costi totali contiene i costi

relativi all'infezione e i costi relativi alle sole azioni attive, quindi consente di comprendere la bontà di un'azione sia in base ai costi sostenuti dalla stessa che per l'effetto generato sull'intero sistema.

$$r_t = \sum_{i=t_i}^{t_f} c_{tot_i}$$

### 5.2.3 Modalità di utilizzo

#### 5.2.3.1 Prima modalità

In una prima modalità di utilizzo del policy maker (PM) con reinforcement learning, l'attivazione è limitata a una singola azione per istante decisionale. L'azione viene eseguita con una frequenza pari a *reductionDuration* istanti temporali, dove *reductionDuration* è il parametro che definisce la durata minima dell'azione selezionata.

In base all'azione selezionata, il PM, prende ogni istante decisionale la decisione di esplorare o sfruttare la strategia migliore secondo quanto salvato in q-table, e attiva l'azione corrispondente.

#### 5.2.3.2 Seconda modalità

Nella seconda modalità di utilizzo si vuole eliminare il vincolo della prima modalità, riguardo al limite di attivazione di una sola azione per volta. In questo caso il PM può attivare più azioni contemporaneamente, ma la frequenza di attivazione delle azioni è unificata ad un unico parametro *t<sub>rd</sub>* (*a\_reductionDuration*) che nei parametri di default è fissato a 15 istanti.

Questa modalità richiede una modifica della struttura della Q-table, la quale viene ampliata per includere anche le sotto-azioni, risultanti dalle combinazioni di attivazione di più azioni contemporaneamente. Di conseguenza, il set di azioni *A* viene esteso e includerà le colonne:

- 1: Nessuna azione attiva;
- 2: Azione  $a_1$  attiva;
- 3: Azione  $a_2$  attiva;
- 4: Azione  $a_3$  attiva;
- 5: Azioni  $a_1$  e  $a_2$  attive;
- 6: Azioni  $a_1$  e  $a_3$  attive;
- 7: Azioni  $a_2$  e  $a_3$  attive;
- 8: Tutte le azioni attive.

Azioni associate ad $A$	$a_1$	$a_2$	$a_3$
1			
2	X		
3		X	
4			X
5	X	X	
6	X		X
7		X	X
8	X	X	X

Il vettore  $A$  ha ora una lunghezza di  $\|A\| = 8$  e di conseguenza la q-table  $Q_{pm}$  per la seconda modalità ha dimensioni  $12 \cdot 8$  con novantasei combinazioni di coppie stato-azioni possibili.

#### 5.2.4 Modifica Q-Table

A seguito di alcune simulazioni è emerso che la Q-table non era in grado di apprendere correttamente le azioni ottimali da intraprendere. Tale fenomeno è stato attribuito all'azione  $a_3$ , caratterizzata da un costo fisso imputato unicamente al primo utilizzo. Nelle simulazioni si osservava infatti un comportamento anomalo: Al primo utilizzo di  $a_3$ , il costo veniva registrato nella Q-table presentava una componente riguardante il costo degli infetti e una relativa al costo delle azioni; tuttavia, al secondo utilizzo la componente relativa al costo delle azioni risultava nulla (assenza del costo fisso dell'azione), causando una riduzione artificiale del valore. Ad esempio con un tasso di apprendimento  $\alpha = 0,5$ , il costo registrato nella Q-table veniva dunque dimezzato al salvataggio del secondo utilizzo di  $a_3$ . Di conseguenza, quando il decisore (policy maker) basava le proprie decisioni sul valore stimato dalla Q-table, si verificava la selezione dell'azione  $a_3$  anche in condizioni in cui non era effettivamente la più vantaggiosa in termini di minimizzazione dei costi complessivi.

In diverse simulazioni è stato possibile osservare che l'azione  $a_3$  veniva attivata in seguito all'utilizzo dell'azione  $a_2$ , sebbene tale scelta non risultasse particolarmente conveniente nella maggior parte dei casi. In tali circostanze, sarebbe stato preferibile proseguire con l'azione  $a_2$  o, alternativamente, sostenere immediatamente il costo fisso di  $a_3$  e continua-

re con quest'ultima, ottimizzando così la strategia complessiva in termini di costi.

È stata quindi necessaria una modifica della Q-table del policy maker  $Q_{pm}$  per includere il costo fisso associato all'azione  $a_3$ , al fine di consentire un corretto apprendimento della strategia ottimale. La modifica ha comportato una distinzione nelle colonne della matrice delle azioni  $A$  relative ad  $a_3$ , separando i casi in cui  $a_3$  viene attivata per la prima volta da quelli in cui è già stata utilizzata. Il vettore  $A$  è stato quindi ulteriormente esteso con l'aggiunta dei seguenti elementi per i casi un cui  $a_3$  è stata già attivata in passato:

- 9: Azione  $a_3$  attiva;
- 10: Azioni  $a_1$  e  $a_3$  attive;
- 11: Azioni  $a_2$  e  $a_3$  attive;
- 12: Tutte le azioni attive.

Azioni associate ad $A$	$a_1$	$a_2$	$a_3$
...			
8			X
9		X	X
10		X	X
11	X	X	X

### 5.2.5 Riepilogo Q-Table

Con le modifiche apportate, la q-table presenta un vettore degli stati  $S$  di lunghezza  $\|S\| = 12$  ed un vettore delle azioni  $A$  con una lunghezza di  $\|A\| = 12$ . La matrice  $Q_{pm}$  ha quindi dimensioni  $12 \cdot 12$  con 144 combinazioni di coppie stato-azioni possibili. Essa presenta quindi la seguente forma:

$Q_{pm}$	$a_1$	$a_2$	$a_3$	...	$a_{10}$	$a_{11}$	$a_{12}$
$s_1$	$Q_{pm}(s_1, a_1)$	$Q_{pm}(s_1, a_2)$	$Q_{pm}(s_1, a_3)$		$Q_{pm}(s_1, a_{10})$	$Q_{pm}(s_1, a_{11})$	$Q_{pm}(s_1, a_{12})$
$s_2$	$Q_{pm}(s_2, a_1)$	$Q_{pm}(s_2, a_2)$	$Q_{pm}(s_2, a_3)$		$Q_{pm}(s_2, a_{10})$	$Q_{pm}(s_2, a_{11})$	$Q_{pm}(s_2, a_{12})$
$s_3$	$Q_{pm}(s_3, a_1)$	$Q_{pm}(s_3, a_2)$	$Q_{pm}(s_3, a_3)$		$Q_{pm}(s_3, a_{10})$	$Q_{pm}(s_3, a_{11})$	$Q_{pm}(s_3, a_{12})$
...				...	...		
$s_{10}$	$Q_{pm}(s_{10}, a_1)$	$Q_{pm}(s_{10}, a_2)$	$Q_{pm}(s_{10}, a_3)$		$Q_{pm}(s_{10}, a_{10})$	$Q_{pm}(s_{10}, a_{11})$	$Q_{pm}(s_{10}, a_{12})$
$s_{11}$	$Q_{pm}(s_{11}, a_1)$	$Q_{pm}(s_{11}, a_2)$	$Q_{pm}(s_{11}, a_3)$		$Q_{pm}(s_{11}, a_{10})$	$Q_{pm}(s_{11}, a_{11})$	$Q_{pm}(s_{11}, a_{12})$
$s_{12}$	$Q_{pm}(s_{12}, a_1)$	$Q_{pm}(s_{12}, a_2)$	$Q_{pm}(s_{12}, a_3)$		$Q_{pm}(s_{12}, a_{10})$	$Q_{pm}(s_{12}, a_{11})$	$Q_{pm}(s_{12}, a_{12})$

### 5.3 Trade-off esplorazione o sfruttamento

Nel modello è stato implementato l'algoritmo  $\varepsilon$ -greedy, il quale consente al policy maker di esplorare inizialmente le azioni in modo casuale (fase esplorativa), per poi passare a una fase di sfruttamento in cui privilegia le azioni con valori più alti (in quanto si tratta di un'ottimizzazione di costo) nella q-table (fase di sfruttamento).

Al fine di incentivare l'esplorazione anche per valori di  $\varepsilon$  ridotti, ogniqualvolta che il policy maker si trova in uno stato in cui il valore massimo della q-table risulti nullo, ossia in combinazioni stato-azione non ancora esplorate, egli seleziona una delle azioni con valore pari a zero. Questo approccio permette di registrare i costi relativi all'azione scelta, sostituendo i valori nulli e fornendo al policy maker una conoscenza più approfondita attraverso una maggiore esplorazione delle possibili azioni.

### 5.4 Decisione del policy maker

#### 5.4.1 Miglior azione all'istante $t$

Il decisore, dotato di capacità di apprendimento per rinforzo (reinforcement learning), implementa le azioni necessarie per contenere il contagio in funzione delle condizioni contingenti. Quando il policy maker sfrutta le conoscenze acquisite (sfruttamento), egli seleziona l'azione ottimale in funzione dell'ottimizzazione del costo totale della simulazione, basandosi sui valori memorizzati nella q-table.

Nel processo di selezione dell'azione ottimale, viene salvato il valore di ogni azione in una variabile temporanea  $c_{at}$ , e viene poi identificato il valore massimo tra le azioni disponibili. La selezione del valore massimo, rispetto al valore minimo, avviene in quanto il valore in q-table sono negativi, quindi l'azione con il valore più alto è quella che minimizza il costo totale della simulazione.

#### 5.4.2 Previsione del prossimo risultato

Utilizzando la una versione modificata della "Bellman Optimality Equation" precedentemente descritta in Sezione 5.2.2, il policy maker allo stato attuale non tiene conto dei risultati futuri. Questo avviene poiché è stata rimossa la componente  $\max_{a'} Q(s', a') - Q(s, a)$  la quale consente di incorporare, ponderata da un fattore  $\gamma$ , una stima del valore di costo atteso in futuro.

Per compensare tale mancanza è stato implementato un sistema che stima lo stato futuro

per mezzo si una seconda q-table  $Q_{pm_{future}}$ . Questa seconda q-table presenta le stesse dimensioni di  $Q_{pm}$  ed è quindi rappresentata da una matrice 12·12. In essa viene registrato il valore  $\Delta_i$ , ovvero la differenza tra il numero di contagiati  $i$  al momento dell'interruzione dell'azione e il numero di contagiati  $i$  al momento della sua attivazione. Se il delta da i due valori è positivo, allora ne consegue che la combinazione di azioni attivate ha aumentato il numero di contagiati nel sistema, se esso è invece negativo, allora tale combinazione ha avuto un impatto positivo sulla simulazione riducendo il numero di agenti infetti  $i$ .

Grazie al valore  $\Delta_i$  è possibile ottere, in base a quanto descritto in Sezione 5.2.1, lo stato futuro  $s_{rl_1}$  in cui si troverà il sistema al termine dell'azione, quindi all'istante  $t = t + t_{rl}$ .

Determinato lo stato futuro, è possibile ricavare il vettore delle azioni  $A$  dalla Q-table  $Q_{pm}$  in corrispondenza di  $Q_{pm}(s = s_{rl_1})$ . Dal vettore  $A$  così ottenuto, si può calcolare il costo che il policy maker dovrà sostenere in futuro, corrispondente al valore massimo tra i costi associati alle azioni presenti in  $A$ . Il valore ottenuto viene salvato in una variabile temporanea denominata  $c_{a_{rl}}$ .

Utilizzando questo metodo si riescono a prevedere i risultati dell'applicazione delle azioni all'istante  $t + t_{rl}$  analizzando l'impatto delle azioni fino all'istante  $t + 2 \cdot t_{rl}$ .

#### 5.4.3 Previsione risultati futuri

È stato in seguito ulteriormente modificato il metodo di previsione dei risultati futuri, al fine di consentire una previsione non solo in base all'attuale e prossima scelta ( $t = t + t_{rl}$ ), ma anche in relazione ai contagi a lungo termine delle successive  $n$  scelte ( $t = t + (n_{rl} + 1) \cdot t_{rl}$ ). Per far ciò è stato implementato un ciclo che esplora tra tutte le possibili combinazioni di azioni attivabili all'istante  $t$ . Per ogni combinazione di azioni è presente un ulteriore ciclo che per ogni azione aggiunge alla variabile  $c_{a_{rl}}$  il costo associato e identifica il prossimo stato nel quale si troverà il sistema. In questo modo la variabile temporanea conterrà il costo che il policy maker sosterrà in futuro in base alla combinazione di azioni analizzate.

**Esempio 8.** *Nel caso venissero considerate le combinazioni di azioni 1234 su  $n_{rl} = 4$  scelte, verranno inseriti in  $c_{a_{rl}}$  i costi delle azioni 1, 2, 3 e 4. Il ciclo inizia aggiungendo a  $c_{a_{rl}}$  i costi di 1 allo stato  $s$  in cui si trova dopo aver applicato  $a_t$  (i cui costi sono contenuti in  $c_{a_t}$ ) ( $c_{a_{rl}} + Q_{pm}(s, 1)$ ). Per mezzo della q-table  $Q_{pm_{future}}$  viene ottenuta la riduzione o l'aumento di agenti infetti a seguito dell'applicazione dell'azione 1. Aggiungendo il valore ottenuto a  $i$  si valuta lo stato futuro in cui il pm si troverà all'istante  $t = (1 + 1) \cdot t_{rl}$ . Il ciclo prosegue quindi con l'aggiunta del costo dell'azione 2 allo stato  $s$  in cui si trova il sistema*

dopo aver applicato  $a_t$  e 1. Il processo si ripete fino ad aver considerato tutte le azioni attivabili all'istante  $t$ .

La previsione tramite la q-table  $Q_{pm_{future}}$ , anziché attraverso la classica “Bellman Optimality Equation”, è stata scelta in quanto consente una maggiore flessibilità e chiarezza nella previsione dei risultati futuri.

L'utilizzo di questa modalità per la previsione degli stati futuri risulta avere una maggior efficienza in caso di simulazioni con seme variabile, poiché vengono esplorate con maggior facilità combinazioni differenti.

Questa modalità verrà impiegata nel calcolo del parametro  $\delta$  durante l'analisi di sensibilità (Sezione 5.6), poiché la previsione dei risultati futuri con  $n_{rl} = 1$  ha dimostrato una maggiore efficienza computazionale rispetto alla previsione con  $n_{rl} = 4$ . Inoltre, la modalità con  $n_{rl} = 0$ , risulta sufficiente nelle restanti analisi, in quanto la comparazione tra le coppie di azioni ( $a_1, a_2$ ) e  $a_3$  permettono di interrompere il contagio entro due istanti decisionali  $t_{rl}$  ( $t + 2 \cdot t_{rl}$ ).

#### 5.4.4 Determinazione dell'azione ottimale

La determinazione dell'azione ottimale da intraprendere all'istante  $t$ , tenendo conto dei benefici o delle complicazioni future, avviene valutando i costi ottenuti dalle q-table. Nello specifico, vengono unite le due variabili precedentemente inizializzate con la seguente relazione:

$$c_{a_{tot}} = c_{a_t} + (\gamma \cdot c_{a_{rl}})$$

In questo modo viene reintegrata la componente  $\gamma \cdot \max_a Q(s_t, a)$  della Bellman Optimality Equation.

L'azione con il valore di somma maggiore sarà quella da perseguire, poiché minimizzerà i costi totali della simulazione.

Nell'Algoritmo 4 è possibile osservare il processo di determinazione del costo totale  $c_{a_{tot}}$  di una qualsiasi azione. L'azione con valore  $c_{a_{tot}}$  maggiore sarà quella selezionata dal policy maker.

---

**Algoritmo 4** Determinazione dell'azione ottimale

---

**Input:** Variabili globali  $gv$ , parametri  $par$ , stato simulazione  $s$

**for**  $a$  in  $Q_{pm}(s)$  **do**

$$c_{at} \leftarrow Q_{pm}(s, a)$$

$$\Delta_i \leftarrow Q_{pm_{future}}(s, a)$$

$$s_{future} \leftarrow get\_state(i + \Delta_i)$$

$$c_{ar_l} \leftarrow max(Q_{pm}(s_{future}))$$

$$c_{atot} \leftarrow c_{at} + c_{ar_l}$$

**end for**

---

## 5.5 Risultati

### 5.5.1 Simulazioni a seme fisso

Nelle seguenti simulazioni viene verificato il corretto funzionamento del policy maker con capacità di apprendimento per rinforzo. In particolare, si intende verificare se il policy maker è in grado di apprendere le azioni ottimali da intraprendere per minimizzare i costi totali della simulazione. La fase di apprendimento e verifica dei risultati si è rivelata la piuttosto laboriosa, poiché ha comportato la correzione di diverse problematiche insite nel codice del modello. Le simulazioni sono state effettuate, se non specificato altrimenti, con un seme fisso pari a  $seed = 2002$  e con parametri default del modello visibili in Sezione 7.1. Ciascuna epoca si compone di 100 simulazioni, dove solamente le ultime 30 vengono prese in analisi in quanto l'algoritmo di  $\varepsilon$ -greedy porta il valore di  $\varepsilon_{RL}$  ad un valore nullo, sfruttando in toto i dati salvati in q-table per la scelta delle azioni.

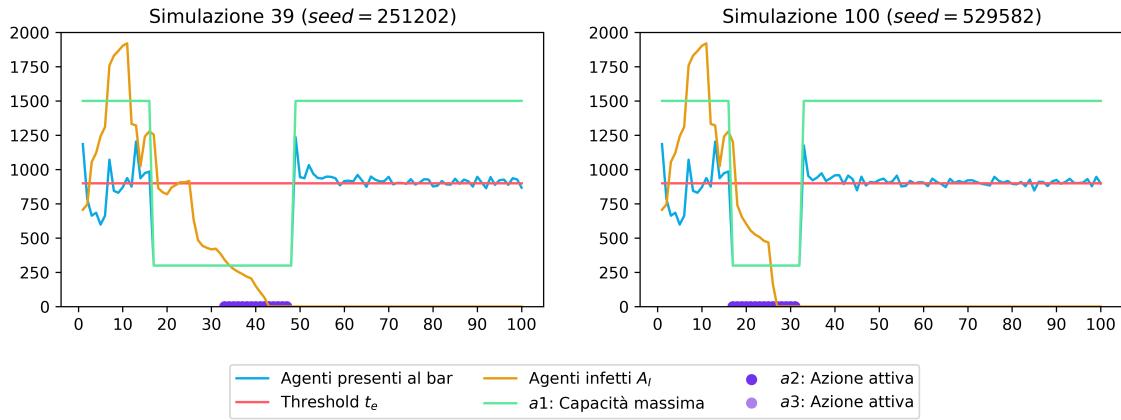


Figura 19: Epoca default comparazione 6, 100 ( $seed = 2002$ )

Viene verificato il funzionamento del policy maker in Figura 19, dove si può notare come all'aumentare del numero di simulazioni venga meglio ottimizzata la funzione obiettivo di costo totale. In figura viene confrontata la simulazione 6 con l'ultima disponibile (la numero 100), quindi teoricamente la miglior in termini di ottimizzazione. Si può notare come il costo totale in simulazione 100 sia stato ridotto del 24,09% rispetto alla simulazione 6, passando da 596.754€ a 453.022€.

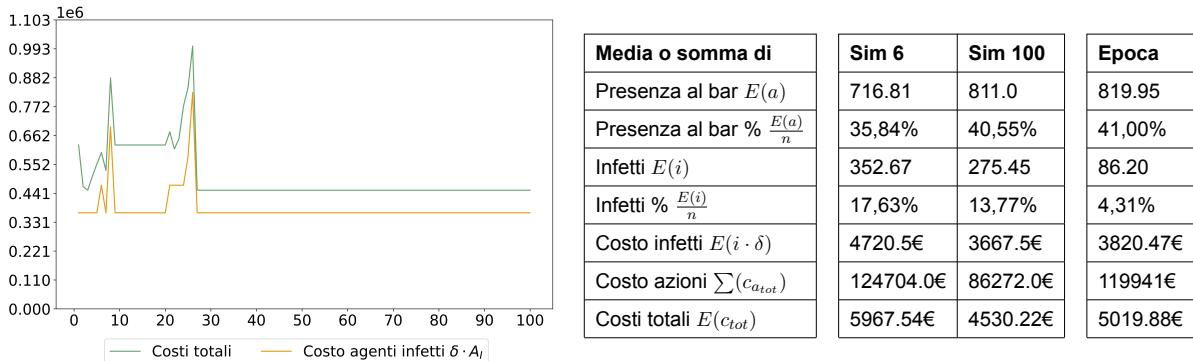


Figura 20: Andamento costi simulazione epoca default ( $seed = 2002$ )

In Figura 5.5.1 si può osservare l'andamento dei costi totali e costi associati agli ingetti medi per ogni simulazione effettuata nell'epoca con parametri default. I costi riescono ad essere ridotti grazie a una miglior gestione del contagio da parte del pm nella simulazione 100. Egli infatti ha appreso che la sola azione  $a_1$  non è sufficiente a risolvere il contagio, così applica fin da subito l'azione  $a_2$ . Il policy maker ha inoltre rilevato che  $a_2$  comporta

un costo variabile e che, riducendo la capacità massima del bar (attivando l'azione  $a1$ ), è possibile diminuire tale costo riducendo il numero di mascherine utilizzate, poiché un numero inferiore di agenti si presenterà al bar a causa del vincolo sulla capacità massima. Tramite le informazioni salvate nella matrice stato-azione  $Q_{pm}$ , il policy maker, ha scartato con successo l'azione  $a3$  in quanto essa comporta un costo fisso di 100.000€ (secondo i parametri default), mentre attivando la coppia di azioni  $a1$   $a2$  si sostengono (in simulazione 100) un costo legato alle azioni  $c_{atot} = 86.272\text{€}$  risparmiando 13.728€.

### 5.5.2 Simulazioni a seme variabile

Di seguito viene esaminato il funzionamento della modalità di reinforcement learning per il policy maker in condizioni in cui il seme è variabile a ogni simulazione. In questo contesto, si verifica la capacità del policy maker di identificare pattern replicabili per la riduzione del contagio su simulazioni caratterizzate da lievi differenze.

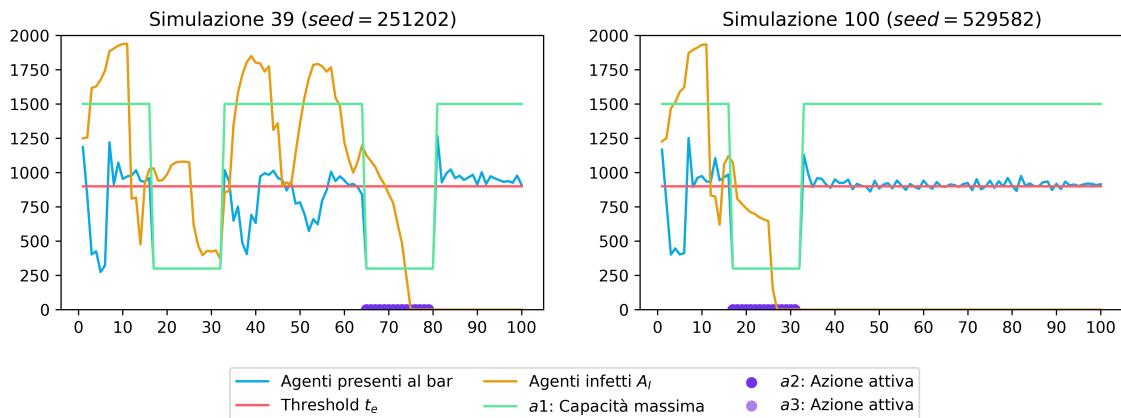


Figura 21: Epoca default comparazione 39, 100

In Figura 21 notiamo come il pm riesca ad identificare la corretta strategia corretto alla simulazione 100, come visto anche nel caso a seme costante. Nella figura è riportata la simulazione 39, in cui il policy maker esplora ampiamente durante la simulazione, aumentando il costo totale ma acquisendo informazioni utili sulle azioni da evitare. Inizialmente, viene attivata l'azione  $a1$ , che non riesce a contenere il contagio. Successivamente, il policy maker tenta di applicare la combinazione nulla, ovvero la combinazione in cui nessuna azione è attiva. Infine, riesce a risolvere il contagio attraverso l'applicazione di una combinazione di  $a1$   $a2$ .

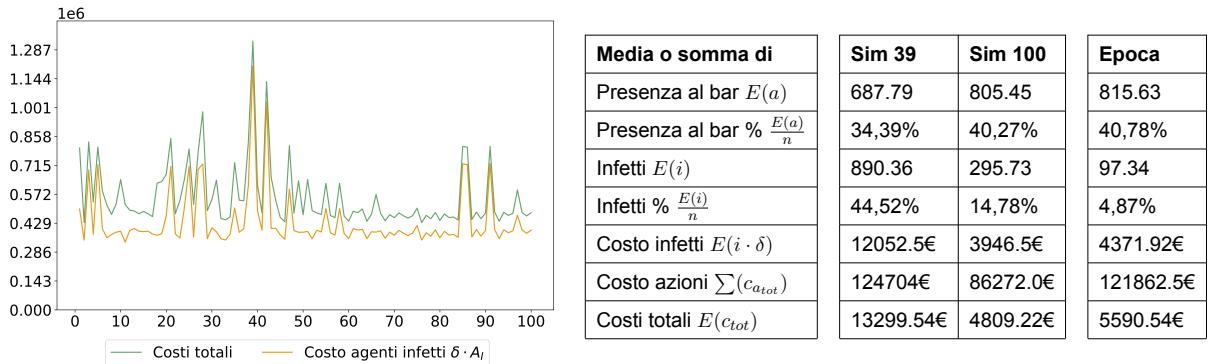


Figura 22: Andamento costi simulazione epoca default ( $seed = 2002$ )

L’andamento dei costi della simulazione (Figura 5.5.2) presenta una natura oscillatoria, determinata dall’aumento della stocasticità tra le diverse simulazioni. In particolare, si può osservare come l’utilizzo di un seme differente possa, per come è stato impostato il modello, far variare il numero di infetti  $i$ , influenzando di conseguenza l’andamento dei costi.

Si può osservare un aumento dei costi verso la fine, corrispondente a valori di  $\varepsilon$  piuttosto elevati, in concomitanza delle simulazioni 84, 85 e 91. In queste simulazioni, la maggiore variabilità del numero di contagi, dovuta all’uso di un seme dinamico, ha portato il policy maker a trovarsi in stati mai esplorati in precedenza. Sebbene il policy maker si trovi in fase di sfruttamento, è stato previsto nel modello che, qualora incontri un valore nullo, ovvero una coppia stato-azione non ancora esplorata, applichi tale azione indipendentemente dalla fase (sfruttamento o esplorazione) in cui si trova. La preposizione all’esplo-razione di coppie stato-azione non ancora affrontate, ed i risultati generati da simulazioni di esplorazione, sono la causa degli innalzamenti di costo a fine epoca.

## 5.6 Analisi di sensibilità

### 5.6.1 Introduzione

In questa sezione verranno analizzati i risultati ottenuti dalle simulazioni col fine di identificare modalità per l’ottimizzazione della funzione obiettivo di costo totale. Verranno proposti vari scenari nel quale si varierà uno o più parametri del modello, al fine di valutare l’impatto di tali variazioni sui costi totali analizzando i comportamenti del pm.

### 5.6.2 Costo $a2$ ( $a2\_cost\_1$ e $a2\_cost\_2$ )

In questa sezione viene analizzato l'impatto di una modifica dei costi associati all'azione  $a2$ . L'azione in questione ha infatti due costi associati ad essa, uno per la mascherina di tipo uno e uno per la mascherina di tipo due. I costi sono definiti dai parametri  $a2\_cost\_1$  e  $a2\_cost\_2$  e sono inizialmente pari a 10€ e 20€ rispettivamente.

Nei risultati che verranno proposti di seguito verrà sempre fatto riferimento al parametro  $a2\_cost\_1$  con  $a2\_cost$ , in quanto  $a2\_cost\_2$  verrà sempre aggiornato con un valore pari al doppio di  $a2\_cost\_1$  ( $a2\_cost\_2 = a2\_cost\_1 \cdot 2$ ).

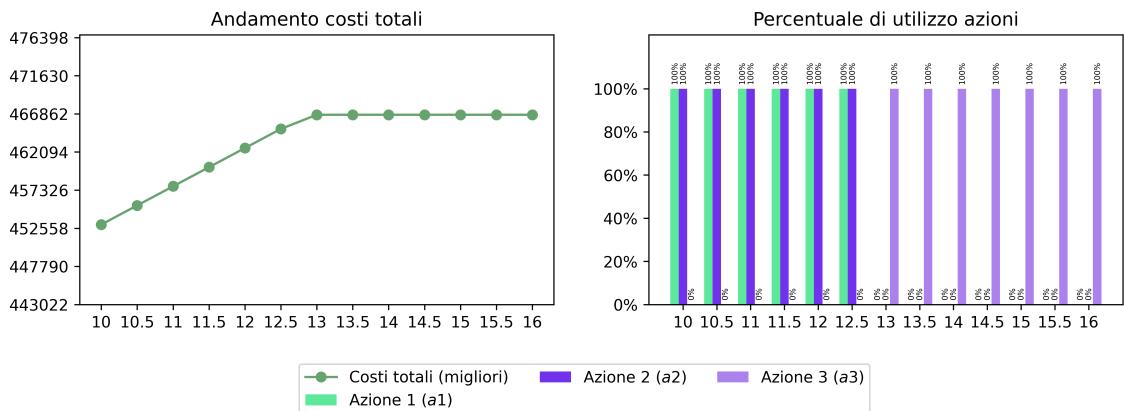


Figura 23: Utilizzo azioni in relazione al costo di  $a2$  ( $seed = 2002$ )

Azione	$a1$	$a2$	$a3$
$a2\_cost < 13\text{€}$	X	X	
$a2\_cost \geq 13\text{€}$			X

In Figura 23 sono mostrati due grafici frutto dei dati ottenuti dalle simulazioni. L'andamento dei costi totali mostra ascisse la variazione del parametro  $a2\_cost$  e in ordinate la media dei costi totali delle cento simulazioni, ovvero:

$$\frac{\sum_{i=1}^{100} c_{tot_i}}{100}$$

La percentuale di utilizzo azioni mostra quante volte è stata utilizzata una determinata strategia in rapporto al numero di simulazioni considerate per ogni epoca. Come descritto in precedenza, il numero di simulazioni considerato è pari a trenta. Pertanto, verrà conteggiato il numero di volte in cui un'azione viene attivata e il totale sarà poi diviso per il numero complessivo di simulazioni (trenta). Il grafico mostrante la percentuale di utilizzo delle azioni, presenterà valori compresi tra zero e uno. Nel caso invece di un seme

costante, i valori potranno essere solamente zero o uno. Ciò avviene in quanto il punto di ottimo per il cambio delle combinazioni di strategie viene identificato in un valore singolo in caso di seme fisso, mentre in un'intorno di tale valore in caso di seme variabile.

Utilizzando il grafico a destra, si può osservare come il policy maker scelga di attivare l'azione  $a_3$  anziché la coppia di azioni del caso base di default  $a_1, a_2$  per valori di  $a_2\_cost$  superiori a 12,5€. Il cambiamento di strategia è intuibile anche dal grafico dell'andamento dei costi delle azioni; infatti, a partire dal valore di 13€ in poi, i costi smettono di aumentare. Questo comportamento si verifica poiché l'incremento di  $a_2\_cost$  aveva un impatto sui costi associati all'azione  $a_2$  e, di conseguenza, sui costi totali. Tuttavia, oltre tale soglia, il policy maker sceglie di non utilizzare l'azione  $a_2$ , rendendo quindi le variazioni del suo costo prive di impatto sul sistema.

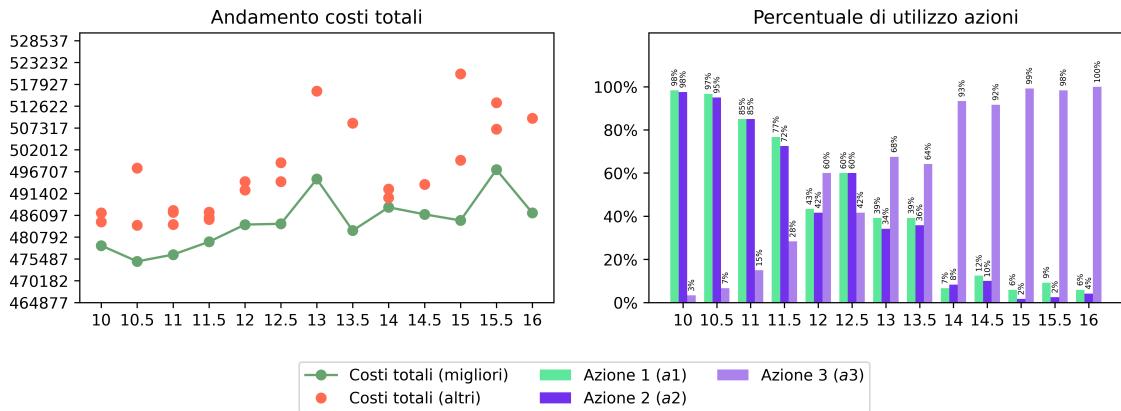


Figura 24: Utilizzo azioni in relazione al costo di  $a_2$  con seme variabile

In Figura 24 è riportata la medesima analisi, questa volta con seme variabile a ogni simulazione (e non fisso a  $seed = 2002$ ). Nel grafico dell'andamento dei costi totali, le epoche con costi totali minori sono evidenziate in verde, mentre le restanti sono indicate in rosso. Dal grafico della percentuale di utilizzo delle azioni, si osserva come il passaggio da una combinazione di azioni a un'altra non sia istantaneo, ma graduale. Tale fenomeno si verifica poiché, avendo ogni simulazione un seme differente, il punto ottimale per il cambio di strategia varia da una simulazione all'altra.

### 5.6.3 Costo $a3$

Analogamente con quanto scritto per il costo di  $a2$  è stato analizzato anche l'impatto del costo di  $a3$  sulle simulazioni. Il costo di  $a3$  è definito dal parametro  $a3\_cost$  e inizialmente è pari a 100.000€.

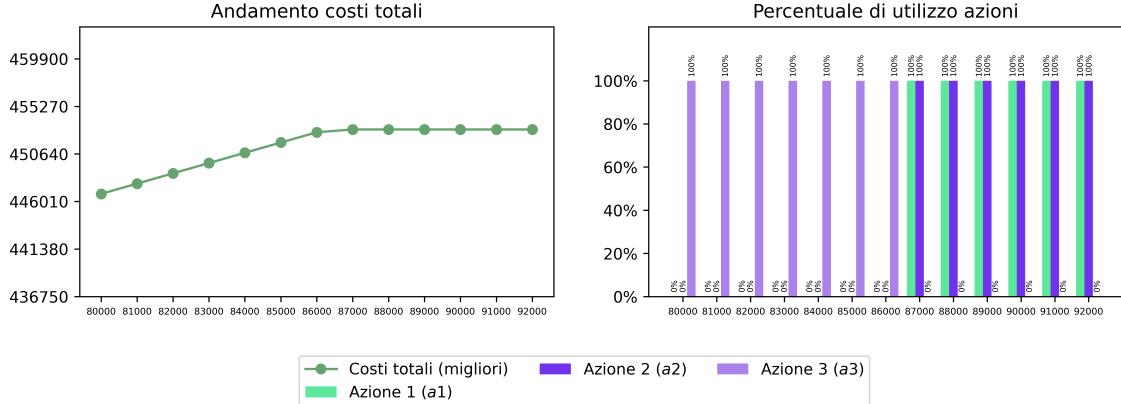


Figura 25: Utilizzo azioni in relazione al costo di  $a3$  ( $seed = 2002$ )

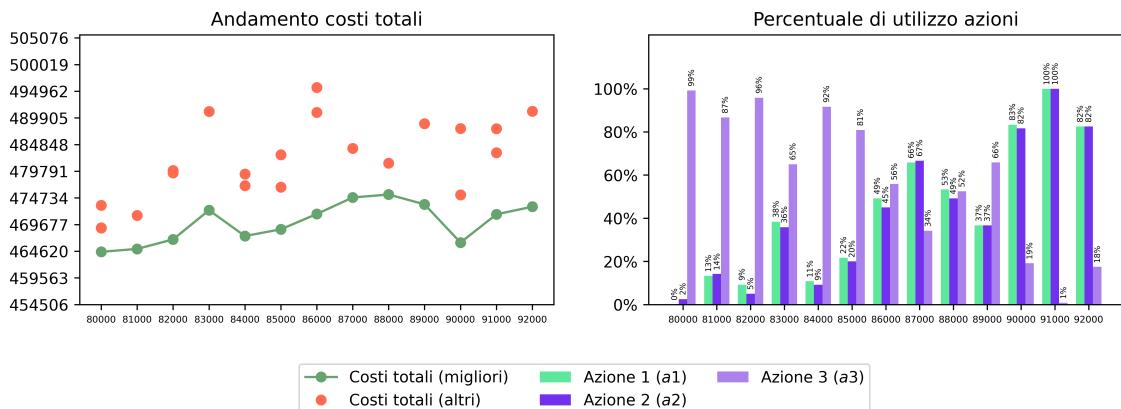


Figura 26: Utilizzo azioni in relazione al costo di  $a3$  con seme variabile

Azione	$a1$	$a2$	$a3$
$a3\_cost < 87.000\text{€}$			X
$a3\_cost \geq 87.000\text{€}$	X	X	

In Figura 25 si può osservare come il policy maker scelga di attivare l'azione  $a3$  in alternativa alla coppia di default  $a1$ ,  $a2$  per valori di  $a3\_cost$  superiori a 87.000€. Dal grafico dei costi delle azioni si nota che, oltre il valore di 87.000€, i costi smettono di aumentare. Questo avviene poiché l'aumento di  $a3\_cost$  influisce sui costi totali fino al momento in cui

il policy maker decide di non usare più l'azione  $a_3$ , rendendo così irrilevanti ulteriori variazioni del suo costo. Il comportamento appena descritto è analogo a quanto osservato per  $a_2$ .

#### 5.6.4 Costo $a_1$

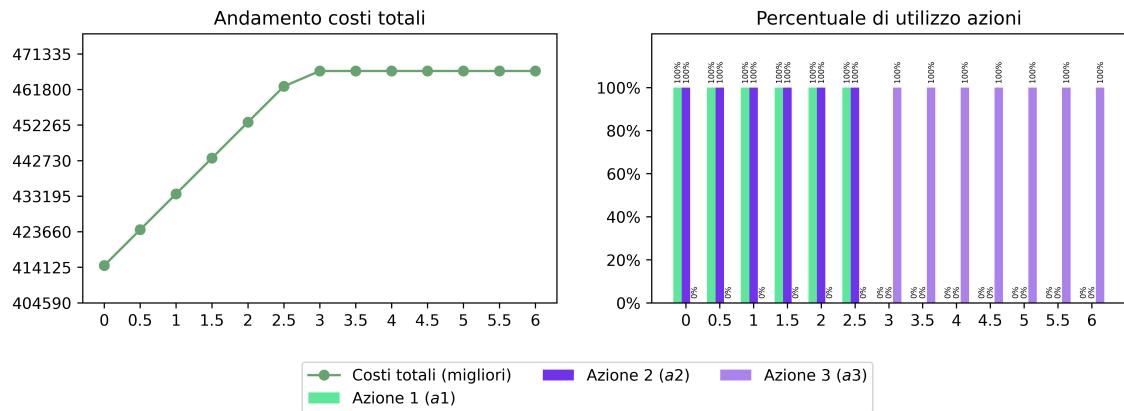


Figura 27: Utilizzo azioni in relazione al costo di  $a_1$  ( $seed = 2002$ )

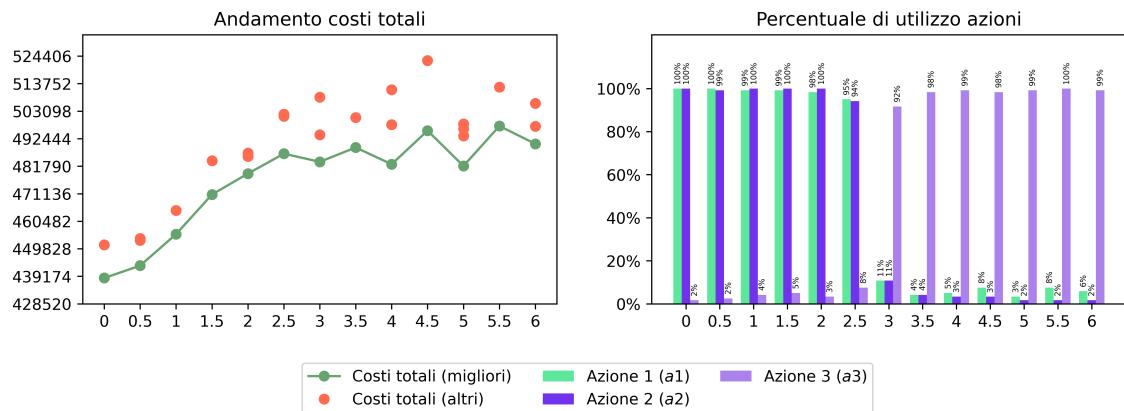


Figura 28: Utilizzo azioni in relazione al costo di  $a_1$  con seme variabile

Azione	$a_1$	$a_2$	$a_3$
$a_1\_cost < 3\text{€}$			X
$a_1\_cost \geq 3\text{€}$	X	X	

Le simulazioni mostrate nelle Figure 27 e 28 mostrano i risultati dell'analisi di sensibilità per il parametro  $a_1\_cost$ , inizialmente fissato a 2€. Si osserva che il policy maker sceglie di attivare l'azione  $a_3$  anziché la combinazione di azioni  $a_1, a_2$  quando  $a_1\_cost$  supera i 3€.

Dal grafico dei costi delle azioni emerge infatti che, oltre questa soglia, i costi totali smettono di aumentare, poiché ulteriori incrementi di  $a1\_cost$  diventano irrilevanti, essendo l'azione  $a1$  non più selezionata dal policy maker.

### 5.6.5 Confronto coppia di azioni $a1, a2$ con l'azione $a3$

Come visto nelle Sezioni 5.6.2, 5.6.3 e 5.6.4 nelle analisi riportate, il policy maker tende ad azionare la coppia di azioni  $(a1, a2)$  oppure l'azione  $a3$ .

La scelta di utilizzare l'azione  $a2$  in combinazione con  $a1$  deriva dal fatto che entrambe presentano un costo variabile, ma con una differenza significativa: il costo unitario di  $a2$  è maggiore di quello di  $a1$ . Per ottimizzare i costi, l'azione  $a1$  viene attivata con il principale scopo di ridurre il costo complessivo di  $a2$ . Infatti,  $a1$  ha un impatto diretto sui costi associati a  $a2$  poiché limita la capacità del bar, riducendo così il numero di agenti presenti al bar  $Att$ . Poiché il costo di  $a2$  dipende dal prodotto tra un costo unitario e il valore di  $Att$ , riducendo  $Att$  si ottiene una riduzione nel costo totale di  $a2$ . Pertanto, il costo complessivo derivante dall'attivazione simultanea delle azioni può essere rappresentato come:

$$c_{a1 \ a2} = c_{a1} + c_{a2} = (a1\_cost \cdot (c_b - (c_b \cdot (1 - r_{a1})))) + ((a2\_cost\_1 \cdot Att_{m1}) + (a2\_cost\_2 \cdot Att_{m2}))$$

Il policy maker predilige la coppia di azioni  $(a1, a2)$  rispetto alla singola azione  $a3$  quando il costo associato alla coppia non supera quello di  $a3$ . Nel caso base, utilizzando i parametri di default, il costo di  $a3$  è fissato a centomila euro ( $a3\_cost = 100.000$ ), quindi le azioni  $(a1, a2)$  vengono selezionate ogni volta che il loro costo combinato  $c_{a1 \ a2}$  è inferiore a questa soglia. In un'analisi di sensibilità su  $a3$ , modificando quindi il valore  $a3\_cost$ , il costo della coppia  $(a1, a2)$  rimarrebbe invariato, e l'azione  $a3$  verrebbe scelta solo se il suo costo scendesse al di sotto di  $c_{a1 \ a2}$ .

### 5.6.6 Costo degli infetti $\delta$

Per effettuare le simulazioni del parametro  $\delta$ , si è utilizzata la previsione dei risultati futuri descritta in Sezione 5.4.3. In particolare, si è scelto di utilizzare  $n_{rl} = 4$  per la previsione, in quanto tale valore ha dimostrato di essere sufficiente per la previsione dei risultati futuri. In questo caso non risultano convenienti valori di  $n_{rl}$  inferiori a quattro, in quanto viene comparata l'interruzione del contagio per mezzo dell'applicazione della coppia  $(a1, a2)$  con l'assenza di qualsiasi intervento strategico. Il valore di  $n_{rl}$  viene identificato considerando

il parametro  $t_{minPm}$ , ovvero il tempo in cui il pm non applica nessuna azione, il parametro  $a\_reductionDuration$  e la durata della simulazione  $t_{max}$ . La relazione tra  $n_{rl}$  e questi parametri è la seguente:

$$n_{rl} + 1 = \left\lfloor \frac{t_{max} - t_{minPm}}{a\_reductionDuration} \right\rfloor = \left\lfloor \frac{100 - 15}{100} \right\rfloor = \lfloor 5,66 \rfloor = 5$$

$$n_{rl} = 5 - 1 = 4$$

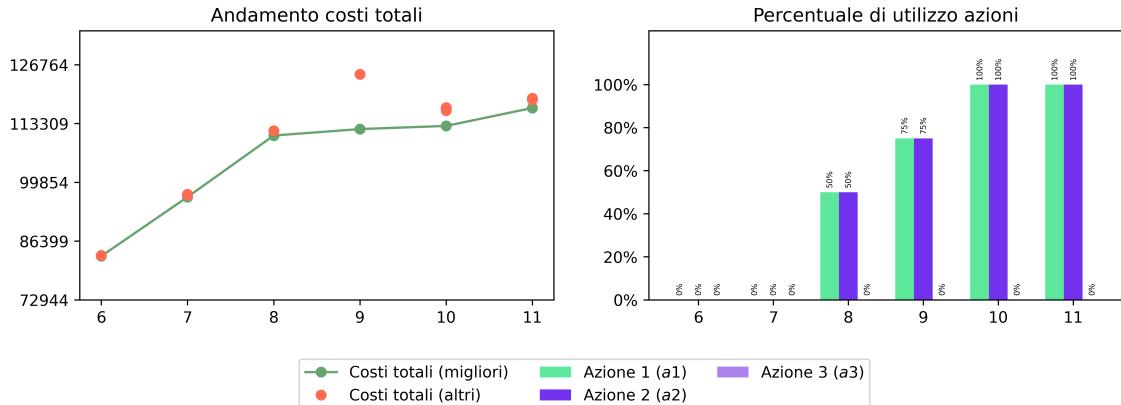


Figura 29: Utilizzo azioni in relazione al costo dei nuovi infetti  $\delta$  con seme variabile

Azione	$a1$	$a2$	$a3$
$\delta < 8\text{€}$			
$\delta \geq 8\text{€}$	X	X	

Nelle epoche mostrate in Figura 29, viene mostrato l'impatto del parametro  $\delta$  sulle simulazioni. Con valori di  $\delta$  inferiori a 8, il policy maker tende a non intraprendere alcuna azione,

consentendo agli agenti di contagiare liberamente altri individui. Al contrario, con valori di  $\delta \geq 8$ , il policy maker inizia a considerare, in alcune simulazioni, l'applicazione della copia di azioni ( $a1, a2$ ). Come discusso in precedenza, questa combinazione rappresenta la strategia ottimale in termini di rapporto costo-efficacia per la riduzione del contagio.

Un comportamento del policy maker di questo tipo è prevedibile, poiché l'agente (pm) non è stato modellato per l'applicazione dell'azione eticamente più appropriata. Infatti, non possiede una particolare consapevolezza o considerazione riguardo all'incremento dei contagi nel sistema, bensì si concentra esclusivamente sull'aumento dei costi, che includono anche quelli derivanti dai contagi stessi. Pertanto, il policy maker sarà disposto a intervenire per fermare il contagio solo quando tale intervento risulti economicamente sostenibile. I punti rossi graficati mostrano le epoche con costi più alti tra le quattro com-

parate. In tutte le epoche, ad eccezione di quella con  $\delta = 9$ , le tecniche di reinforcement learning implementate sono riuscite a raggiungere l'ottimo minimizzando gli errori.

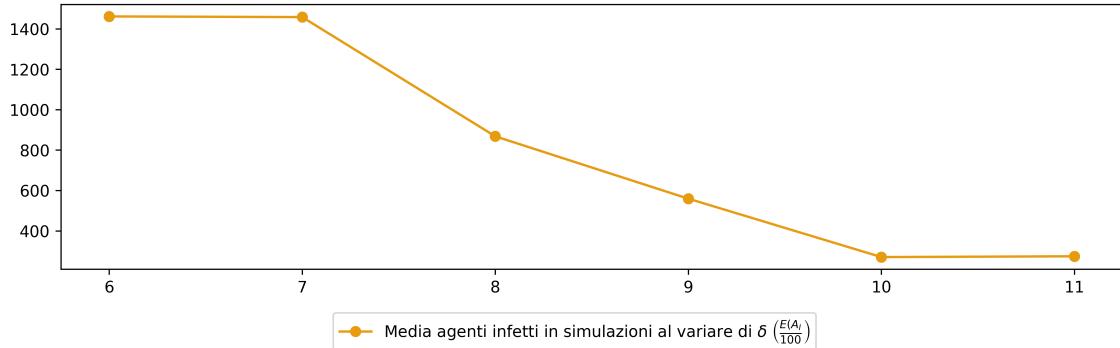


Figura 30: Andamento del contagio al variare di  $\delta$  con seme variabile

In Figura 30 è possibile osservare l'andamento del contagio al variare di  $\delta$ . Più precisamente nel grafico viene mostrata la media degli agenti infetti per ogni epoca analizzata ( $E(A_I)$ ). Si può notare come in media il numero di infetti cali drasticamente applicando le azioni  $a_1 a_2$ , quindi in corrispondenza di valori  $\delta \geq 8$ .

### 5.6.7 Parametri durata contagio $t_i, t_r$

In questa sezione si vuole verificare il funzionamento del policy maker variando i valori di  $t_i$  e  $t_r$ . I valori del caso base sono pari a 10 e 3 rispettivamente. I due parametri regolano la durata del contagio, tempo nel quale un agente  $a$  rimane nello stato  $A_I$ , e il tempo di recupero, tempo nel quale un agente  $a$  rimane nello stato  $A_R$ . Ciò significa che l'agente permarrà infetto per tredici istanti di tempo  $t$  e in fase di recupero per tre. Quando l'agente  $a$  entra a far parte del set di agenti infetti  $A_I$  non potrà essere infettato nuovamente per tredici giorni ( $t_i + t_r = 13$ ).

Nelle seguenti simulazioni si modificano i suddetti parametri impostandoli pari a:

- $t_i = 25$
- $t_r = 10$

Si intende analizzare i comportamenti del policy maker in relazione alle variazioni del parametro  $a\_reductionDuration$ , che indica la durata delle azioni (se la simulazione è in modalità di reinforcement learning di tipo due, come in questo caso). Come precedentemente descritto, il policy maker ha visibilità su due istanti decisionali, consentendogli di prendere la decisione corretta per l'istante decisionale attuale ( $t$ ) e per il suc-

cessivo ( $t + a\_reductionDuration$ ). Nel caso base, tale parametro è pari a quindici ( $a\_reductionDuration = 15$ ), un valore superiore al periodo durante il quale un agente infetto non può essere reinfettato. Aumentando i parametri  $t_i$  e  $t_r$  a valori la cui somma è pari a 35, il policy maker prenderebbe decisioni all'istante  $t$  per i successivi  $a\_reductionDuration \cdot 2 = 30$  istanti, che risulterebbero inferiori alla durata complessiva di  $t_i + t_r$  ( $30 \leq 35$ ). Questa problematica può essere ovviata aumentando il parametro  $a\_reductionDuration$  a trentacinque ( $a\_reductionDuration \rightarrow 35$ ), in modo da consentire al policy maker di prendere decisioni verificando l'impatto delle proprie azioni. Di seguito si analizzano i seguenti due casi:

- $a\_reductionDuration = 15 \mid a\_reductionDuration < (t_i + t_r)$
- $a\_reductionDuration = 35 \mid a\_reductionDuration \geq (t_i + t_r)$

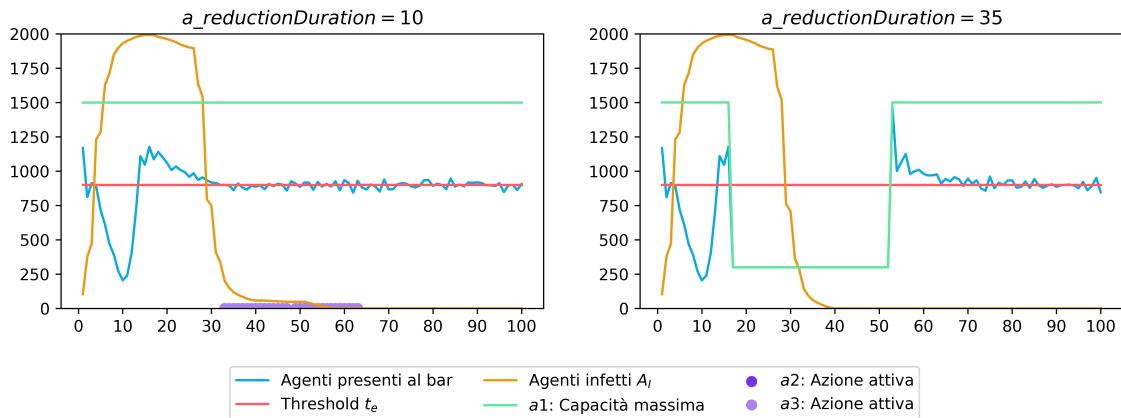


Figura 31: Aumento  $t_i$  e  $t_r$  ( $seed = 2002$ )

In Figura 31 si possono osservare i comportamenti del policy maker in entrambe le casistiche. Le due immagini mostrate mostrano le simulazioni numero cento delle due epoche. Nel caso in cui  $a\_reductionDuration = 15$ , il policy maker cerca di ridurre il numero di agenti infetti entro 15 istanti, applicando l'azione  $a3$ , che si rivela la più efficace per contenere il contagio in questa configurazione ma in linea di massima la più costosa. All'aumentare di  $a\_reductionDuration$ , il policy maker applica invece l'azione  $a1$ , poiché, con i parametri  $t_i$  e  $t_r$  modificati, essa risulta sufficiente per contenere il contagio, consentendo un risparmio di 22.078€. In particolare, la simulazione con  $a\_reductionDuration = 10$  comporta costi totali pari a 392.200€, mentre la simulazione con  $a\_reductionDuration = 35$  comporta costi totali pari a 370.122€.

## 6 Conclusione

L'utilizzo degli ABM consente di ricreare ambienti complessi che possono descrivere in modo accurato la realtà. L'applicazione di algoritmi di reinforcement learning per la ricerca dell'ottimo in tali modelli permette, qualora essi siano sufficientemente rappresentativi, di individuare strategie ottimali applicabili anche a contesti reali. Poiché un ABM può descrivere scenari sia attuali sia proiettati nel futuro, l'impiego del reinforcement learning all'interno di tali modelli offre la possibilità di identificare scelte ottimali per scenari futuri, fornendo così supporto decisionale per situazioni che potrebbero verificarsi nel mondo reale.

Un lato negativo riscontrato sin dagli albori con i modelli ad agenti è il notevole impiego di risorse necessario per le simulazioni. Aggiungendo a modelli ABM l'apprendimento per rinforzo si aumenta ulteriormente l'utilizzo di risorse. I risultati presentati in questa tesi hanno richiesto in media 25 minuti per epoca, per un totale di 100 epoche per analisi. Inoltre le analisi di sensibilità a seme fisso sono state effettuate generando un'epoca per ogni parametro modificato, mentre le simulazioni a seme variabile quattro epoche per ciascun parametro. Questo esempio evidenzia come il raggiungimento di risultati significativi richieda un impegno considerevole in termini di tempo e risorse.

Tali problematiche risultano essere attualmente un prezzo da pagare per l'utilizzo di queste tecnologie. Tuttavia, l'ABM si conferma una tecnica di modellazione estremamente versatile e adattabile a diversi contesti applicativi. Il modello presentato in questa tesi, pur essendo stato progettato per simulare una dinamica di contagio, può essere generalizzato e reinterpretato anche in un'ottica gestionale. Di seguito vengono proposti alcuni esempi di potenziali applicazioni gestionali, derivanti da un'interpretazione alternativa della struttura implementata:

### **Esempio 9. Strategie di marketing**

*Ipotizziamo l'applicazione del modello in ambito di gestione delle strategie di marketing di un prodotto x. In questo caso tra gli n agenti ci potrebbe un sottogruppo che viene identificato come target del nostro prodotto x. In relazione al modello presentato il sotto gruppo di agenti potrebbe essere identificato negli agenti che si presentano al bar Att. A questo punto potremmo utilizzare il modello SIRS per identificare gli agenti S come coloro che non hanno ancora acquistato il prodotto x, I coloro che hanno acquistato da poco il prodotto e infine R come coloro che hanno acquistato x ma che sono più*

interessati ad utilizzarlo. A discrezione dello scenario da replicare si può fare pensare di riportare gli agenti allo stato  $S$  dopo determinati istanti di tempo  $t$ . Infine il policy maker rappresenterebbe l'addetto alla campagna marketing che applicherebbe le azioni  $a_1$ ,  $a_2$  e  $a_3$  per identificare la strategia migliore per la vendita del prodotto  $x$ .

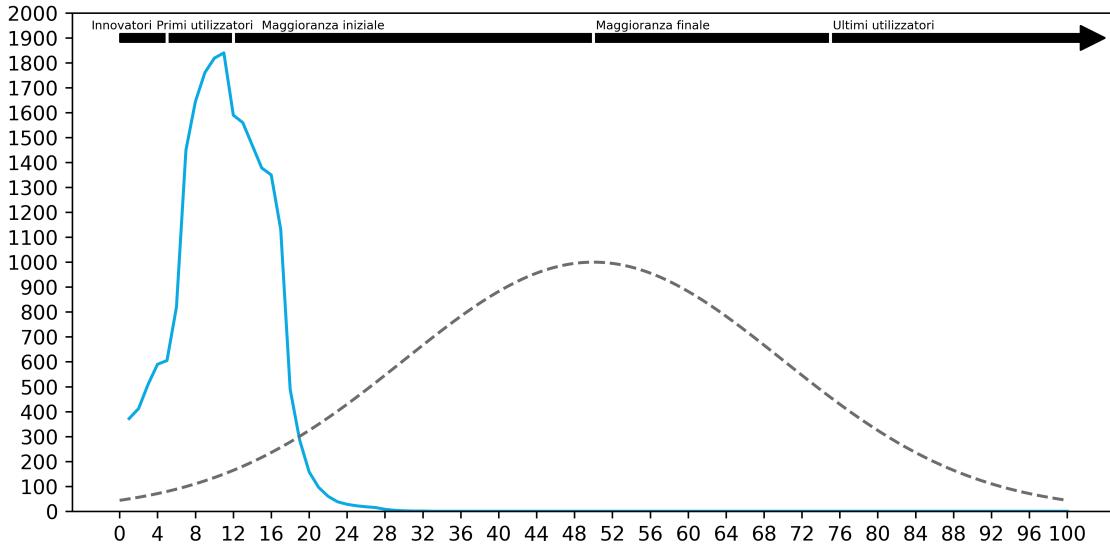


Figura 32: Simulazione SIR ( $seed = 2002$ )

In Figura 32 viene mostrata la similarità tra il modello di diffusione shark fin proposto da Larry Downes e Paul Nunes in [20]. In grigio è possibile osservare la curva a campana del modello di diffusione dell'innovazione classico, mentre in blu la shark fin ottenuta per mezzo dei risultati ottenuti dal modello in modalità SIR (Sezione 4.2.1.3). In questo caso potremmo quindi interpretare la curva degli agenti  $A_I$  come la curva di diffusione del prodotto  $x$ .

#### Esempio 10. Applicazione logistica

Consideriamo gli agenti come un insieme di  $n$  automezzi (autoarticolati o autocarri) incaricati di trasportare merci fino a un transit point (il "bar"). Gli automezzi che arrivano a questo transit point costituiscono il sottoinsieme  $Att$ . Ciascun automezzo può trovarsi in uno stato SIRS: lo stato  $S$  rappresenta gli autocarri vuoti o pieni in attesa di essere caricati o scaricati; lo stato  $I$  identifica gli autocarri attualmente in fase di carico o scarico presso il transit point; infine, lo stato  $R$  indica gli automezzi in attesa o che sono pronti a partire per effettuare il trasporto verso la destinazione successiva. Il policy maker può essere visto come il responsabile della logistica, incaricato di gestire le partenze e le attese degli autocarri, con l'obiettivo di ottimizzare il flusso di mezzi al transit point, con

*l'obiettivo di minimizzare il tempo associato ad  $R$  che gli automezzi trascorrono nel transit point. Le azioni  $a_1$ ,  $a_2$ , e  $a_3$  permetterebbero una migliore organizzazione del transit point, facilitando un'allocazione più efficiente degli automezzi. Il gestore del transit point si avvarrebbe della di un modello simile a quello mostrato in questa tesi per identificare le strategie migliori per la gestione del flusso di automezzi.*

#### **Esempio 11. Gestione delle scorte di un magazzino**

*Ipotizziamo di avere un insieme di  $n$  merci che possono entrare o uscire da un magazzino. Le merci che all'istante  $t$  risiedono all'interno del magazzino fanno parte di  $Att$ . Le scorte nel magazzino possono appartenere allo stato  $S$  se sono ancora materie prime,  $I$  se sono in fase di lavorazione (semilavorati) e  $R$  se si tratta di prodotti finiti. Il policy maker potrebbe essere identificato come il responsabile del magazzino, incaricato di gestire le merci in magazzino attuando approcci di gestione a scorta. Le azioni che il responsabile attuerebbe consentirebbero al magazzino mantenere  $Att$  al di sopra di una soglia necessaria al reparto produttivo. In questo caso egli potrebbe applicare il modello del lotto economico (EOQ) ( $a_1$ ) oppure il modello dell'intervallo fisso di riordino ( $a_2$ ). Si potrebbero avere così altre azioni per ogni variazione dei modelli  $a_1$  e  $a_2$ .*

Inoltre, il modello può esser facilmente scalato cambiando alcuni parametri. Aumentando ad esempio il numero di agenti  $n$  ed il numero di istanti osservati  $t_{max}$  si potrebbe cambiare il soggetto della simulazione, a fronte di un aumento delle risorse necessarie per le simulazioni.

#### **Esempio 12. Gestione epidemie su scala maggiore**

*Modulando il parametro  $n$  e  $c_b$  potremmo generare simulazioni in cui il bar si trasformi in un'intera nazione, regione o città. In questo modo il valore di  $Att$  indicherebbe il numero di agenti all'interno del sistema modellato. Il policy maker potrebbe essere identificato come il responsabile della sanità pubblica, incaricato di gestire l'epidemia in corso tramite l'applicazione delle strategie  $a_1$ ,  $a_2$  e  $a_3$ .*

## Bibliografia

- [1] F. Bertolotti and L. Mari, “Agent-based modeling: un’analisi critica della bibliografia e delle relazioni con la teoria generale dei sistemi,” *Università Cattaneo Research Reports*, vol. 3, 2018.
- [2] Y. Bar-Yam, *Dynamics Of Complex Systems*. Addison-Wesley, 1997.
- [3] A. F. Siegenfeld and Y. Bar-Yam, “An introduction to complex systems science and its applications,” *Complexity*, vol. 2020, 2020.
- [4] J. A. T. Thomas M. Cover, *Elements of Information Theory*. John Wiley and Sons, 2012.
- [5] E. Bonabeau, “Agent-based modeling: Methods and techniques for simulating human systems,” *Proceedings of the national academy of sciences*, vol. 99, pp. 7280–7287, 2002.
- [6] C. M. Macal and M. J. North, “Tutorial on agent-based modeling and simulation,” in *Proceedings of the Winter Simulation Conference, 2005*. IEEE, 2005.
- [7] J. Castiglione, M. Bradley, and J. Griebe, *Activity-based travel demand models: A primer*, 2015.
- [8] J. M. Epstein and R. Axtell, *Growing artificial societies: social science from the bottom up*. Brookings Institution Press, 1996.
- [9] C. Castelfranchi, “Guarantees for autonomy in cognitive agent architecture,” in *International Workshop on Agent Theories, Architectures, and Languages*. Springer, 1994, pp. 56–70.
- [10] N. R. Jennings and M. Wooldridge, “Applying agent technology,” *Applied Artificial Intelligence an International Journal*, vol. 9, pp. 357–369, 1995.
- [11] G. V. e. a. Bobashev, “A hybrid epidemic model: combining the advantages of agent-based and equation-based approaches,” in *2007 winter simulation conference*. IEEE, 2007, pp. 1532–1537.
- [12] V. e. a. Grimm, “A standard protocol for describing individual-based and agent-based models,” *Ecological modelling*, vol. 198, pp. 115–126, 2006.

- [13] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [14] J. Clifton and E. Laber, “Q-learning: Theory and applications,” *Annual Review of Statistics and Its Application*, vol. 7, pp. 279–301, 2020.
- [15] E. B. e. a. Laber, “Optimal treatment allocations in space and time for on-line control of an emerging infectious disease,” *Journal of the Royal Statistical Society Series C: Applied Statistics*, vol. 67, pp. 743–789, 2018.
- [16] M. Labonne. (2022) Q-learning for beginners. [Online]. Available: <https://towardsdatascience.com/q-learning-for-beginners-2837b777741>
- [17] W. B. Arthur, “Inductive reasoning and bounded rationality,” *The American economic review*, vol. 84, pp. 406–411, 1994.
- [18] J. Veisdal. (2020) The el farol bar problem. [Online]. Available: <https://www.cantorsparadise.org/the-el-farol-bar-problem-a60205dd3f86>
- [19] F. Bertolotti, N. Kadera, L. Pasquino, and L. Mari, “Exploring epidemiological dynamics in a social dilemma,” in *Proceedings of the French Regional Conference on Complex Systems, May 29-31, 2024, Montpellier, France*. FRCCS, 2024, pp. 18–33.
- [20] L. Downes and P. Nunes, “Finding your company’s second act,” *Harvard Business Review*, vol. 2018, pp. 98–107, 2018.

## 7 Appendice

### 7.1 Parametri presenti nel modello

All'interno delle parentesi quadre è stato inserito il valore default attribuito ai seguenti parametri:

#### Parametri della simulazione

- $t_{max}$  (**max\_days**): [100] Numero di giorni (o istanti  $t$ ) per ogni simulazione.
- $n$  (**n\_persons**): [2000] Rappresenta il numero di persone per ogni simulazione.

#### Parametri sociali

- $c_b$  (**capacity**): [1500] Questo numero intero rappresenta la capacità massima del bar (è significativo se `respect_the_max`: bool = True).
- $t_e$  (**threshold**): [0,6] Questa soglia viene utilizzata per determinare se un agente andrà al bar o meno a seconda della sua strategia. Si tratta del threshold per la determinazione della presenza al bar di El Farol.
- **respect\_the\_max**: [True] Questo booleano rappresenta se la capacità della bar sarà rispettata o meno.

#### Strategie degli agenti

- **strategyOne**: [0,25] Percentuale sul totale degli agenti che seguiranno la strategia uno per l'ottenimento della prima decisione riguardo alla presenza al bar. La prima strategia è totalmente randomica.
- **strategyTwo**: [1 - strategyOne = 0,75] Percentuale sul totale degli agenti che seguiranno la strategia due per l'ottenimento della prima decisione riguardo alla presenza al bar. La seconda strategia è calcolata parzialmente con una regressione lineare del vettore memoria contenente le precedenti strategie degli agenti.
- **useRegrFrom**: [5] Indica il giorno dal quale gli agenti che seguono la strategia due potranno utilizzare una regressione lineare, in quanto il vettore memoria sarà sufficientemente popolato.

- **useRegrFor:** [0,5] Dell'output totale della strategia definita dall'agente ogni settimana, il valore definito dalla regressione lineare delle precedenti influisce su una percentuale definita dal parametro.

## Parametri epidemiologici

- $t_c$  (**infection\_threshold**): [0,4] Un altro agente diventa contagioso se il suo livello contagio  $c_j$  è maggiore del *contagious\_threshold*. Quindi se un agente ha un livello di contagio inferiore a  $t_c$  non potrà più infettare altri agenti.
- $t_s$  (**infection\_thresholdNotPresent**): [0,8] Implica che un agente con un livello di contagio  $c_j$  maggiore di  $t_s$ , indipendentemente dal valore di della strategia, non si presenterà al bar in quanto i sintomi dell'infezione sono troppo elevati.
- $t_i$  (**infection\_duration**): [10] Identifica la durata in giorni dell'infezione.
- $t_r$  (**infection\_cantStartUntil**): [3] Dopo che un agente guarisce da un'infezione, il valore identifica quanto tempo un agente è immune ad una nuova infezione.
- $\alpha$  (**alpha**): [0,2] Peso che varia il numero di nuovi infetti per agente.
- **infection\_randomness:** [0,05] Altera il livello di contagio aggiungendo una componente stocastica che spazia tra *-infection\_randomness* e *+infection\_randomness*.
- $\delta$  (**delta**): [150] Costo per ogni nuovo infetto
- $r_i(r)$ : [0] Ricavo per ogni agente, viene utilizzato in relazione al numero dei guariti ad ogni istante di tempo.
- **num\_infected\_persons:** [100] Identifica il numero di persone contagiose ad inizio simulazione.
- **infected\_person\_starting\_day\_min:** [-7] Indica il minimo giorno in cui l'agente che ad inizio simulazione è infetto inizia il proprio contagio, è necessario esso sia maggiore di  $t_i$ .
- **infection\_generatesResistance:** [True] Abilita il valore precedente, per l'agente deve attendere  $t_r$  giorni, dopo essere guarito, per essere infettato nuovamente

- **people\_memory\_weight\_arr:** [[0,5; 0,2; 0,1]] Pesi relativi assegnati alla memoria delle precedenti strategie salvate nel vettore memoria. In questo caso, l'ultimo valore inciderà per un 50% sul valore finale della strategia, il penultimo 20%, il terzultimo 10% ed il rimanente 20% verrà ripartito tra tutti gli altri valori presenti nel vettore memoria.
- **regression\_type:** [1] Grado della regressione effettuata con np.polyfit (1 = regressione lineare).
- **infection\_randomness:** [0,25] Altera il livello di contagio per un valore randomico che spazia tra -infection\_randomness e +infection\_randomness.
- **$\delta$  (delta):** [150] Costo per ogni nuovo infetto
- **$r_i(r)$ :** [0] Ricavo per ogni agente, viene utilizzato in relazione al numero dei guariti ad ogni istante di tempo.

### Parametri del Policy Maker

- **enablePM:** [True] Abilita il Policy Maker nella simulazione.
- **enableA1 :** [True] Abilita la strategia A1 del PM. Ovvero l'azione per cui viene ridotta la capacità massima del bar.
- **enableA2 :** [True] Abilita la strategia A2 del PM. Ovvero l'azione per cui viene imposto l'utilizzo di mascherine all'interno del bar.
- **enableA3 :** [True] Abilita la strategia A3 del PM. Ovvero l'azione per cui viene effettuato un test sul livello di contagio all'entrata del bar.
- **enable\_at\_least\_one\_A :** [False] Limita il PM ad abilitare una sola azione, ha effetto quando il reinforcement learning è disabilitato o nella modalità 2.

### Parametri Azioni

Tutte le azioni posseggono un parametro intitolato *reductionDuration* che rappresenta la durata dell'azione, ed un parametro *InfectedThreshold* che viene utilizzato per il policy maker in modalità base (senza reinforcement learning) ed indica la soglia di attivazione della strategia.

## Parametri Azione 1

- $r_{a1}$  (**a1\_reductionPerc**): [0,8] È la percentuale di riduzione della capacità massima della bar nel momento in cui A1 è attiva.
- $c_{a1}$  (**a1\_cost**): [2] Rappresenta il costo dell'azione A1, esso viene moltiplicato per il numero di infetti totali ogni istante in cui l'azione rimane attiva.
- **a1\_reductionDuration** : [25] Durata dell'azione A1 per la modalità uno del reinforcement learning o reinforcement learning disattivato.
- **a1\_InfectedThreshold** : [0,91] Per le simulazioni con reinforcement learning disattivato viene utilizzato per definire quando attivare la strategia A1, ovvero quando il numero di infetti totali supera  $n \cdot a1\_InfectedThreshold$ .

## Parametri Azione 2

- $m1_{a2}$  (**a2\_faceMask1Agents**): [0,65] Percentuale di agenti che indosseranno la mascherina di tipo 1.
- $m2_{a2}$  (**a2\_faceMask2Agents**): [0,3] Percentuale di agenti che indosseranno la mascherina di tipo 2.
- $m0_{a2}$  (**a2\_faceMask0Agents**): [0,05] Percentuale di agenti che indosseranno la mascherina di tipo 0, ovvero coloro che riusciranno ad entrare al bar senza l'ausilio di una mascherina.
- $mp1_{a2}$  (**a2\_faceMask1Perc**): [0,3775] Aumento del threshold di contagio  $t_c$  per gli agenti che utilizzano la mascherina di tipo 1.
- $mp2_{a2}$  (**a2\_faceMask2Perc**): [0,5] Aumento del threshold di contagio  $t_c$  per gli agenti che utilizzano la mascherina di tipo 2.
- $c_{a2_1}$  (**a2\_cost\_1**): [10] Rappresenta il costo dell'azione A2, esso viene moltiplicato per la capienza del bar ogni istante in cui l'azione rimane attiva. Questo costo viene utilizzato per le mascherine di tipo 1.
- $c_{a2_2}$  (**a2\_cost\_2**): [20] Parametro di costo il cui utilizzo è il medesimo rispetto a  $a2\_cost\_1$ , con la differenza che viene utilizzato per le mascherine di tipo 2.

- **a2\_reductionDuration** : [15] Durata dell'azione A2 per la modalità uno del reinforcement learning o reinforcement learning disattivato.
- **a2\_InfectedThreshold** : [0,85] Per le simulazioni con reinforcement learning disattivato viene utilizzato per definire quando attivare la strategia A2, ovvero quando il numero di infetti totali supera  $n \cdot a1\_InfectedThreshold$ .

### Parametri Azione 3

- **a3\_testFailUnder**: [0,4] Percentuale di errore del test.
- $c_{a3}$  (**a3\_cost**): [100.000] Rappresenta il costo dell'azione A3, esso rappresenta un costo fisso e viene contabilizzato solo la prima volta che viene seguita A3.
- **a3\_reductionDuration** : [15] Durata dell'azione A2 per la modalità uno del reinforcement learning o reinforcement learning disattivato.
- **a3\_InfectedThreshold** : [0,3] Per le simulazioni con reinforcement learning disattivato viene utilizzato per definire quando attivare la strategia A3, ovvero quando il numero di infetti totali supera  $n \cdot a1\_InfectedThreshold$ .

### Reinforcement Learning PM

- **enableRL**: [True] Abilita il reinforcement learning per il Policy Maker.
- $\varepsilon_{RL}$  (**epsilon\_RL**): [0,2] Percentuale di esplorazione del PM, al decrescere di  $\varepsilon_{RL}$  il PM esplorerà meno affidandosi maggiormente a già quanto salvato nella q-table. Nelle simulazioni ad epoche viene applicato l'epsilon greedy algorithm.
- $\alpha_{RL}$  (**alpha\_RL**): [0,3] Percentuale per cui viene aggiornato il valore salvato in q-table, con  $\alpha_{RL} = 1$  viene rimpiazzato il valore precedente salvato in q-table.
- $t_{minPM}$  (**RL\_PM\_t\_min**): [15] Tempo minimo per il quale il PM non si attiverà, lasciando diffondere il contagio ad inizio simulazione. Il policy maker entrerà in azione solamente dopo  $RL\_PM\_t\_mingiorni$ .
- $t_{rd}$  **a\_reductionDuration**: [15] Tempo di attivazione (reductionDuration) attribuito a tutte le strategie nel caso di  $RL\_mode = 2$ .
- **RL\_mode**: [2] Definisce la modalità di reinforcement learning, la modalità 1 ha come ipotesi la possibilità di attivare una azione per volta, mentre la modalità 2 consente

l'attivazione di più strategie simultaneamente unificandone il tempo di attivazione (reductionDuration).

- **infection\_slope\_regr\_len:** [5] Per definire lo stato nel quale si trova il PM, viene effettuata una regressione lineare degli ultimi infection\_slope\_regr\_len giorni ed estrapolata la pendenza dello storico delle infezioni.

## **Ringraziamenti**

Colgo questa occasione per esprimere la mia più sincera gratitudine a tutte le persone che mi hanno accompagnato e sostenuto durante questo percorso universitario.

Innanzitutto, un sentito ringraziamento ai miei genitori, che mi hanno permesso di intraprendere questa esperienza accademica, supportandomi e guidandomi nelle mie scelte e decisioni. Un grazie speciale va anche ai miei familiari per il costante supporto e incoraggiamento che non mi hanno mai fatto mancare.

A Sofia, che mi è stata accanto nei momenti di maggiore tensione, offrendo il suo sostegno e la sua pazienza, va la mia più profonda riconoscenza.

Ringrazio amici e compagni universitari che con la loro compagnia hanno reso questi anni più leggeri e ricchi di momenti di condivisione e svago.

Un particolare ringraziamento va ai docenti che hanno contribuito alla mia crescita accademica, in modo speciale al Professor Bertolotti, per la sua guida preziosa durante la stesura di questo elaborato e per il supporto ricevuto lungo tutto il mio percorso formativo.

Pur concludendosi questa fase del mio cammino, guardo con entusiasmo al futuro e sono determinato a proseguire gli studi, portando sempre con me il bagaglio di esperienze e insegnamenti raccolti in questi tre anni.