

# **NXN-IMITATE-IF- BETTER-NOISE- PAYOFF-TO-USE**

---

**Giada Maurizzi**

**Tecniche e Metodi di Simulazione**

GEPID - Governance e Politiche dell'Innovazione Digitale

# Il modello

L'obiettivo è estendere il modello **nxn-imitate-if-better-noise** aggiungendo due caratteristiche che ampliano il modello:

- La possibilità di utilizzare gli **expected payoff**.
- La possibilità di modellare altre **decision rules** oltre a quella dell'imitazione se migliore.

## Aggiungiamo 3 parametri:

### PAYOFF-TO-USE

Questo parametro specifica il tipo di payoff che gli agenti utilizzano in ogni tick.

### DECISION RULE

Questo parametro determina la regola decisionale che gli agenti seguono per aggiornare le proprie strategie.

### PARAMETRO M

Si tratta del parametro che controlla l'intensità della selezione nella regola decisionale diretta-positiva-proporzionale-m.

## Prima estensione

# Implementare diversi metodi di calcolo del payoff

Come calcolo il payoff delle strategie?

**play-with-one-random-agent**

Gli agenti giocano con un altro agente  
scelto a caso

**use-strategy-expected-payoff**

Gli agenti utilizzano il payoff atteso della loro  
strategia

# EXPECTED vs OBSERVED PAYOFF

Consideriamo la matrice di payoff per il gioco "sasso, carta, forbice"



## OBSERVED PAYOFF

Il payoff osservato è il **guadagno effettivo** che un agente ottiene giocando contro un altro agente scelto a caso.

Se un agente gioca contro un altro agente scelto a caso, il payoff osservato **dipenderà dalla strategia dell'avversario**.

$$\text{payoff\_realizzato} = \text{payoff}(s_{\text{giocatore}}, s_{\text{altro}})$$

Ad esempio, se un agente che gioca Sasso incontra un agente che gioca Carta, il payoff osservato sarà -1.



## EXPECTED PAYOFF

L'expected payoff per una strategia è calcolato come la **media dei payoffs** che un agente otterrebbe **giocando con tutti gli altri agenti nella popolazione**. Questo valore tiene conto delle **probabilità associate a ciascuna strategia** degli avversari.

$$\text{payoff\_atteso}(s) = \sum_{s'} (\text{payoff}(s, s') \times P(s'))$$

L'expected payoff viene calcolato considerando le probabilità di incontro con gli altri agenti che giocano S, C o F.

$$E(S) = pS \cdot 0 + pC \cdot 1 + pF \cdot (-1)$$

```

class Player(Agent):

    def __init__(self, unique_id, strategy, model):
        self.unique_id = unique_id
        self.model = model
        self.strategy = strategy
        self.strategy_after_revision = strategy
        self.payoff = 0

    def update_payoff(self):
        self.payoff = self.model.payoff_to_use(self)

    def update_strategy_after_revision(self):
        if random.random() < self.model.prob_revision:
            if random.random() < self.model.noise:
                self.strategy_after_revision = random.randint(0, self.model.n_of_strategies - 1)
            else:
                others = [p for p in self.model.players if p.unique_id != self.unique_id]
                if others:
                    observed_player = self.model.random.choice(others)
                    if observed_player.payoff > self.payoff:
                        self.strategy_after_revision = observed_player.strategy
        else:
            self.strategy_after_revision = self.strategy

    def update_strategy(self):
        self.strategy = self.strategy_after_revision

```

La funzione **payoff\_to\_use** permette di confrontare due modi di valutare il successo:

- Expected: contro distribuzione dell'intera popolazione (teorico).
- Random: contro un avversario casuale (più realistico).

**expected\_payoff** → calcola il payoff atteso come media pesata sulla distribuzione delle strategie nella popolazione.

**realized\_payoff** → calcola il payoff effettivo contro un giocatore selezionato a caso.

- 1.Ogni agente ha una strategia e un guadagno (payoff).
- 2.Può decidere di cambiare strategia, osservando altri.
- 3.Se cambia:
  - può scegliere a caso (rumore)
  - oppure può imitare un altro agente che guadagna più di lui

Alla fine, adotta la nuova strategia scelta.

```

def payoff_to_use(self, player):
    if self.payoff_method == "expected":
        return self.expected_payoff(player.strategy)
    elif self.payoff_method == "random":
        return self.realized_payoff(player)
    else:
        raise ValueError(f"Metodo di guadagno non valido: {self.payoff_method}")

def expected_payoff(self, strategy):
    counts = [sum(1 for p in self.players if p.strategy == s) for s in range(self.n_of_strategies)]
    total = sum(counts)
    if total == 0:
        return 0
    probs = [c / total for c in counts]
    return sum(self.payoff_matrix[strategy][s] * p for s, p in enumerate(probs))

def realized_payoff(self, player):
    others = [p for p in self.players if p.unique_id != player.unique_id]
    if not others:
        return 0
    other = self.random.choice(others)
    return self.payoff_matrix[player.strategy][other.strategy]

```

```

def step(self):
    for player in self.players:
        player.update_payoff()

    for player in self.players:
        player.update_strategy_after_revision()

    strategy_changes = 0
    for idx, player in enumerate(self.players):
        if player.strategy_after_revision != player.strategy:
            strategy_changes += 1
            player.update_strategy()
    self.total_strategy_changes += strategy_changes

    self.update_strategy_expected_payoffs()

    self.data_collector.collect(self)

    strategy_counts = [0] * self.n_of_strategies
    for player in self.players:
        strategy_counts[player.strategy] += 1

    strategy_frequencies = [count / self.num_agents for count in strategy_counts]
    self.strategy_history.append(strategy_frequencies)

    current_snapshot = [p.strategy for p in self.players]
    if self.equilibrium_reached_step is None and current_snapshot == self.last_strategy_snapshot:
        self.equilibrium_reached_step = len(self.strategy_history)
    self.last_strategy_snapshot = current_snapshot

```

- Calcola il payoff attuale di ogni agente.
- Ogni agente decide se e come cambiare strategia (con imitazione e/o noise).
- Viene applicata la nuova strategia.
- Raccoglie dati sulle strategie e payoff (per grafici e analisi).
- Se le strategie non cambiano da un passo al successivo → equilibrio raggiunto

# Ricapitolando

## metodo random payoff observed

- ◆ Il payoff dell'agente si basa su un'**interazione casuale** con un altro agente.
- ◆ È più realistico (una singola esperienza), ma anche più rumoroso/stocastico.
- ◆ Rende la dinamica più soggetta a **fluttuazioni**, quindi più lenta o instabile nel convergere a un equilibrio.

## metodo expected expected payoff

- ◆ Il payoff è calcolato rispetto alla **distribuzione dell'intera popolazione**.
- ◆ È una stima più accurata, meno influenzata da rumore stocastico.
- ◆ Porta spesso a una **dinamica più regolare e prevedibile**.

# Simulazione iniziale: Confronto tra metodi di payoff

**Qual è l'effetto dei diversi metodi di payoff  
sull'evoluzione delle strategie?**

**Parametri:**

**Numero di agenti:** 300

**Distribuzione iniziale equa:** [100, 100, 100]

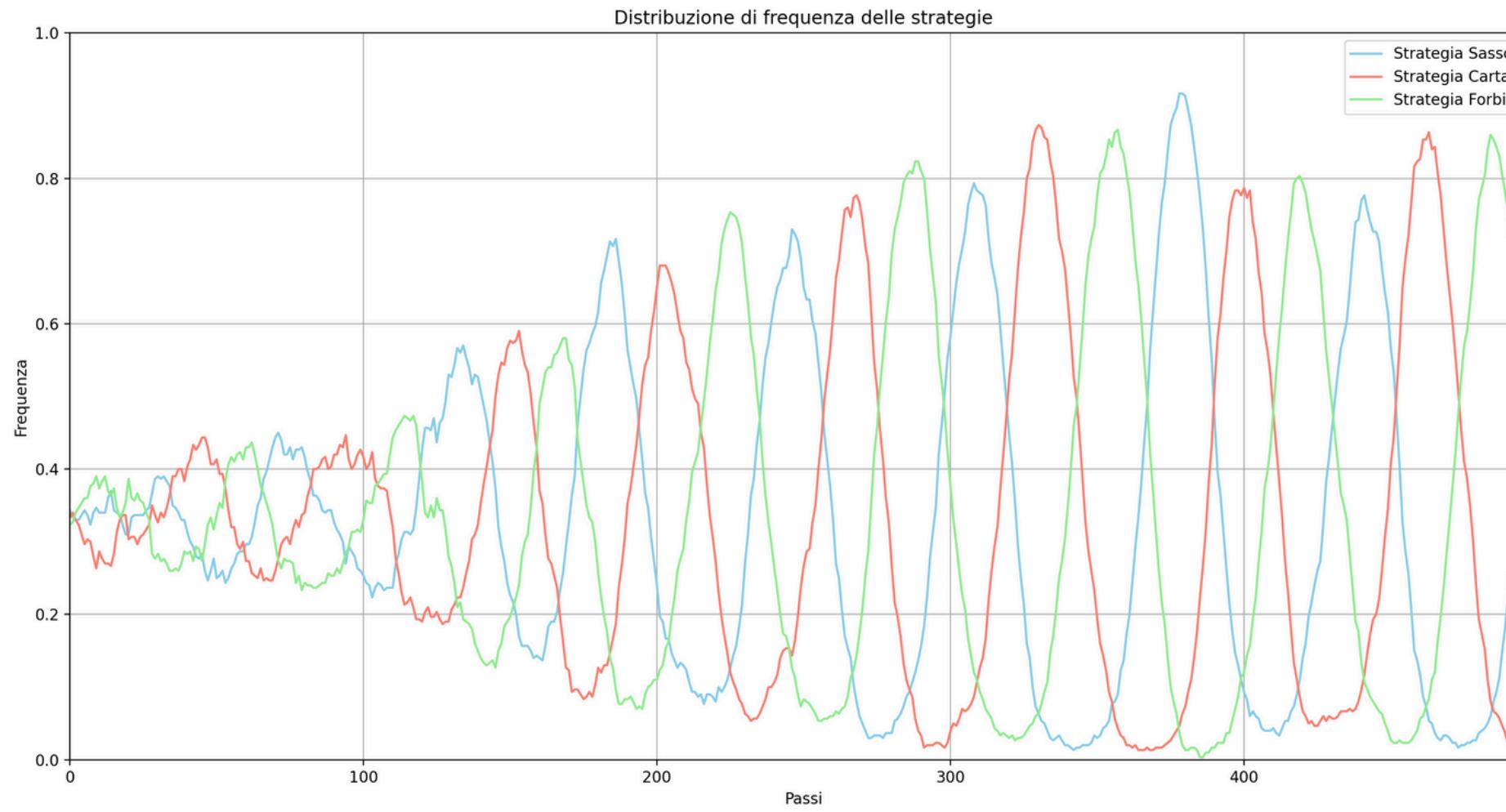
**Matrice dei payoff** = ([[0, -1, 1],  
[1, 0, -1],  
[-1, 1, 0]])

**Noise** = 0.01

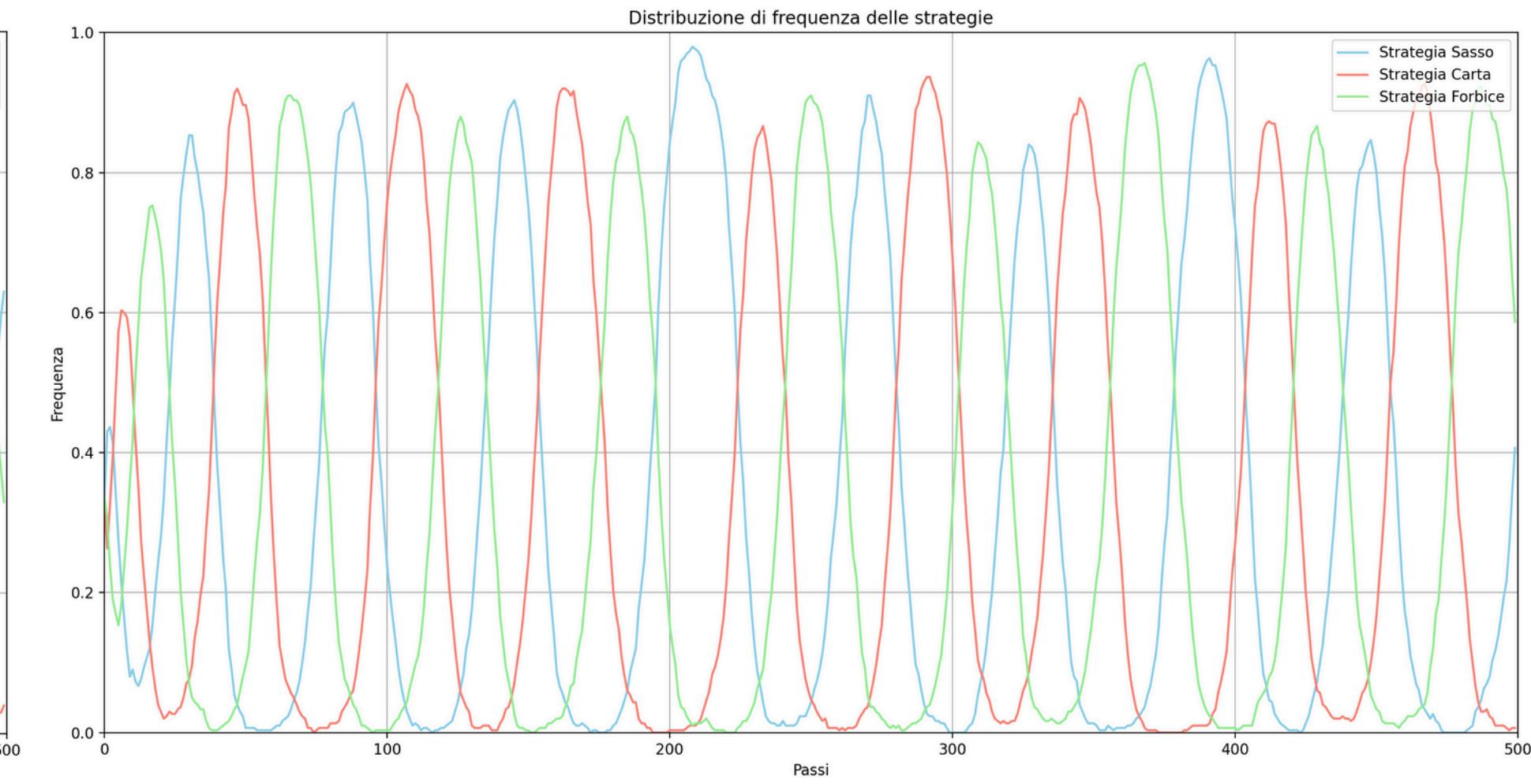
**Prob\_revision** = 0.3

**Steps** = 500

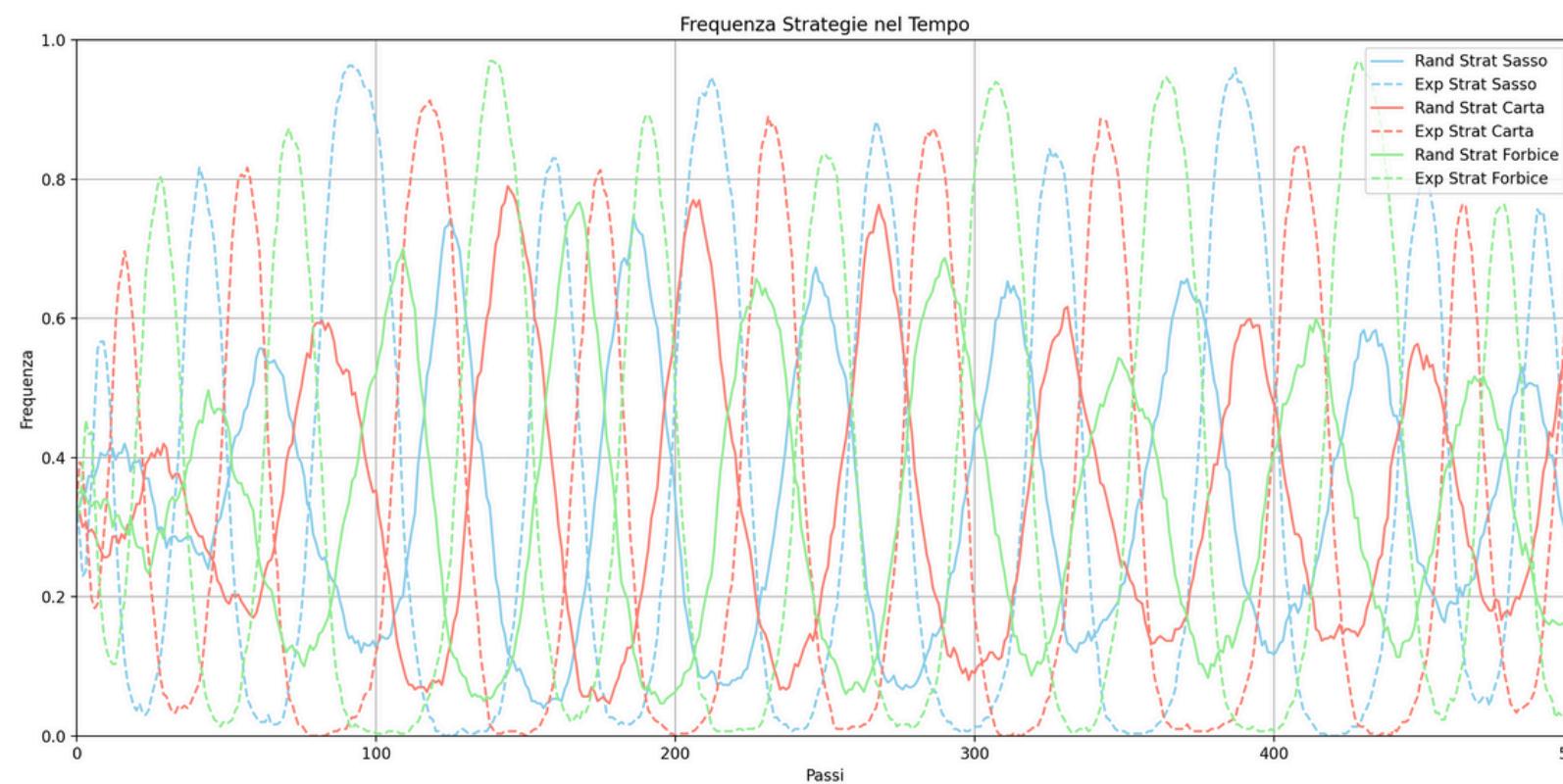
**Evoluzione delle strategie nel tempo - Metodo di Guadagno: random**



**Evoluzione delle strategie nel tempo - Metodo di Guadagno: expected**



**Confronto Evoluzione Strategie  
Metodo Random vs Expected**



# Statistiche Finali: Metodo Random vs Expected

## Metodo Random

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	-0.342	0.529	79	0.263	1	-1
Carta	0.167	0.532	168	0.56	1	-1
Forbice	0.302	0.664	53	0.177	1	-1
--- Totali / Medi ---						
Cambi strategia / step	18.52					
Tempo medio a equilibrio	Mai raggiunto					

## Metodo Expected

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	-0.633	0.0	73	0.243	-0.633	-0.633
Carta	0.205	0.068	215	0.717	0.287	-0.633
Forbice	0.173	0.131	12	0.04	0.347	-0.633
--- Totali / Medi ---						
Cambi strategia / step	16.686					
Tempo medio a equilibrio	142					

# Osservazioni



## CONFRONTO

- Entrambe le distribuzioni hanno un **andamento sinusoidale** anche se il metodo random raggiunge più tardi questa stabilità.
- **Nessuna strategia domina.**
- Le frequenze nel metodo expected sono più alte mentre la varianza tra payoff è nettamente minore.
- I cambi di strategia sono maggiori nel metodo random.
- Il metodo expected ha favorito una convergenza più rapida verso un equilibrio. Al contrario, il metodo random ha mostrato oscillazioni più marcate, con maggiore variabilità nei payoffs e frequenti cambi strategici.
- Questo è coerente col fatto che l'expected payoff fornisce una stima media sull'intera popolazione, portando a **scelte più “razionali”** e stabili, mentre il metodo random introduce più volatilità decisionale.

# Considerazioni

- In giochi dove nessuna strategia domina stabilmente (come Sasso, Carta e Forbice), è normale osservare andamenti oscillanti o sinusoidali delle frequenze strategiche.
- Questo succede perché appena una strategia diventa più frequente, una contro-strategia guadagna vantaggio, e la popolazione si sposta ciclicamente.

***Ci sono situazioni che modificano questo andamento?  
In quale tipo di payoff-to-use prima?***

# Simulazione 2: Come varia la stabilità della strategia all'aumentare delle probabilità di revisione

Un'alta probabilità di revisione favorisce una maggiore diversità strategica o induce la convergenza verso una strategia dominante?

**Parametri:**

**Numero di agenti:** 300

**Distribuzione iniziale equa:** [100, 100, 100]

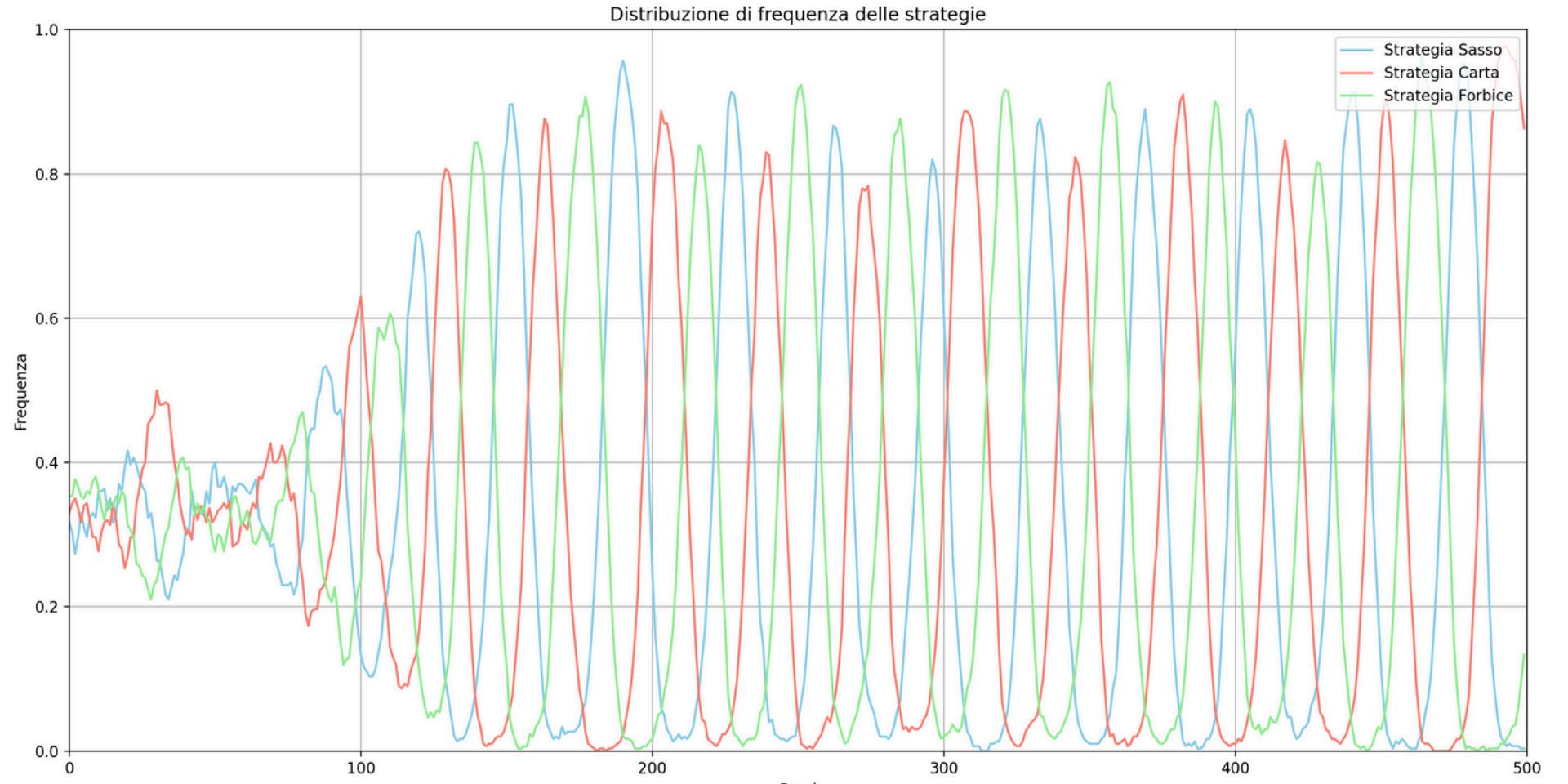
**Matrice dei payoff** = ([[0, -1, 1],  
[1, 0, -1],  
[-1, 1, 0]])

**Noise** = 0.01

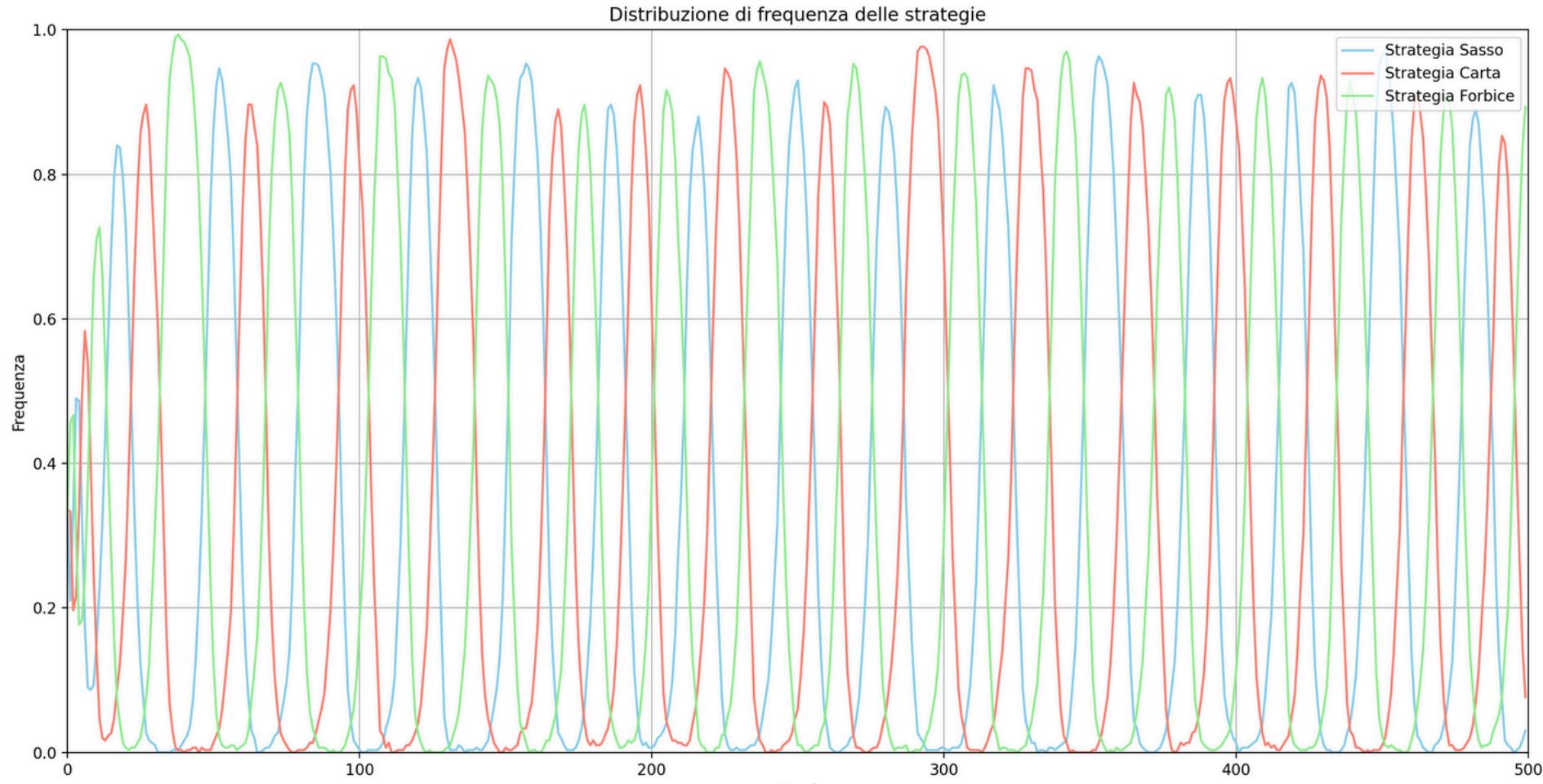
**Prob\_revision** = 0.6 → AUMENTIAMO IL VALORE

**Steps** = 500

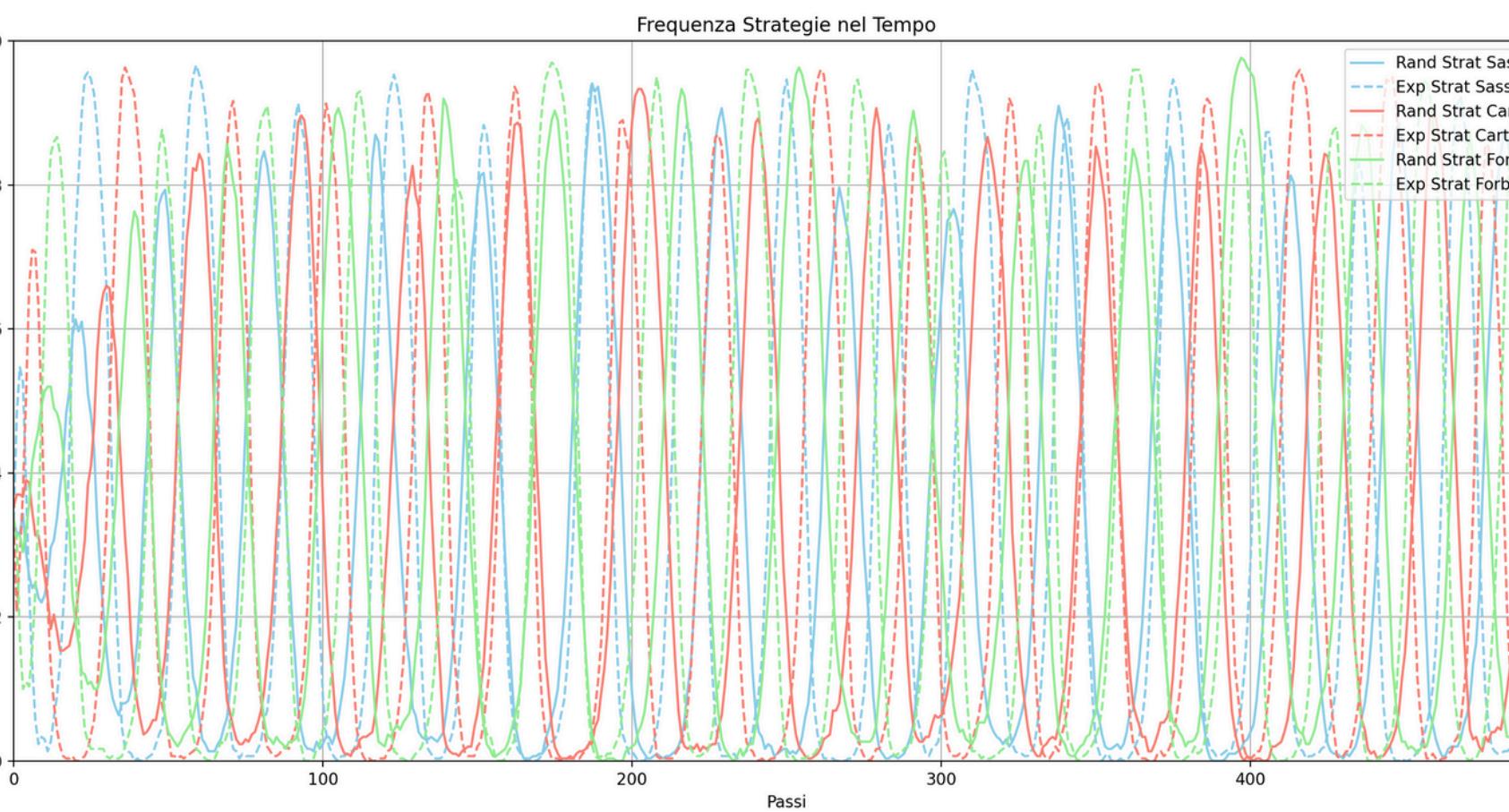
Evoluzione delle strategie nel tempo - Metodo di Guadagno: random



Evoluzione delle strategie nel tempo - Metodo di Guadagno: expected



Confronto Evoluzione Strategie  
Metodo Random vs Expected



## Statistiche Finali: Metodo Random vs Expected

### Metodo Random

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	0.333	0.889	3	0.01	1	-1
Carta	-0.205	0.174	190	0.633	1	-1
Forbice	0.346	0.507	107	0.357	1	-1
--- Totali / Medi ---						
Cambi strategia / step	28.16					
Tempo medio a equilibrio	Mai raggiunto					

### Metodo Expected

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	-0.099	0.004	242	0.807	-0.093	-0.773
Carta	0.476	0.222	54	0.18	0.867	-0.093
Forbice	-0.433	0.116	4	0.013	-0.093	-0.773
--- Totali / Medi ---						
Cambi strategia / step	29.284					
Tempo medio a equilibrio	Mai raggiunto					

# Osservazioni



## CONFRONTO

- L'aumento della probabilità di revisione, porta **gli agenti a cambiare strategia spesso**.
- Numero di cambi strategici per passo alto in entrambi i metodi.
- Maggiore **instabilità** nelle frequenze delle strategie nel tempo.
- Le **fluttuazioni** sono **maggiori** rispetto a prima.

# Simulazione 3: Introduzione del rumore

**Qual è l'effetto del rumore sul comportamento strategico e sulla convergenza del sistema?**

**Parametri:**

**Numero di agenti:** 300

**Distribuzione iniziale equa:** [100, 100, 100]

**Matrice dei payoff** = ([[0, -1, 1],  
[1, 0, -1],  
[-1, 1, 0]])

**Noise** = 0.6 → AUMENTIAMO IL VALORE

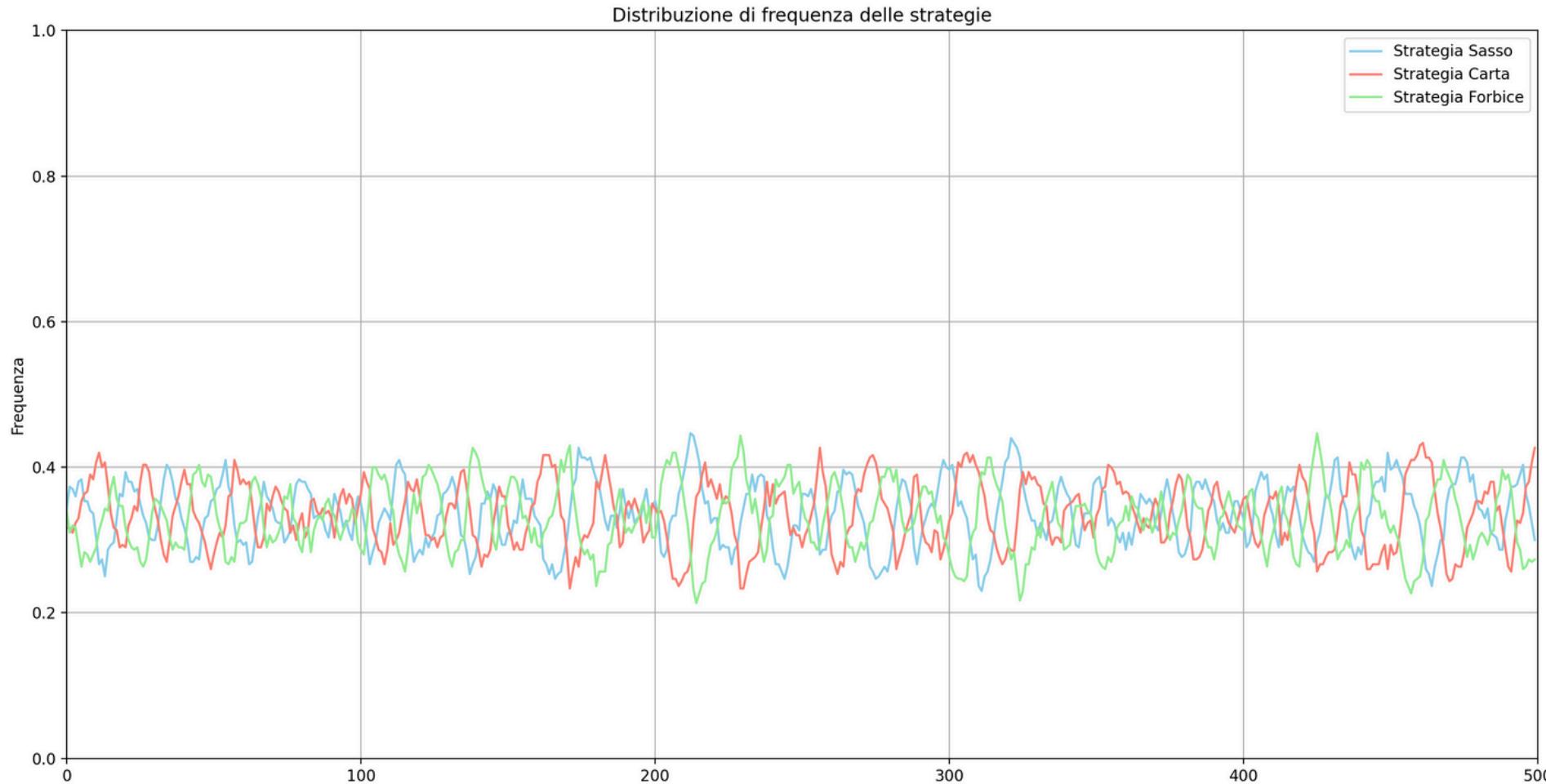
**Prob\_revision** = 0.2 → VALORE MEDIO

**Steps** = 500

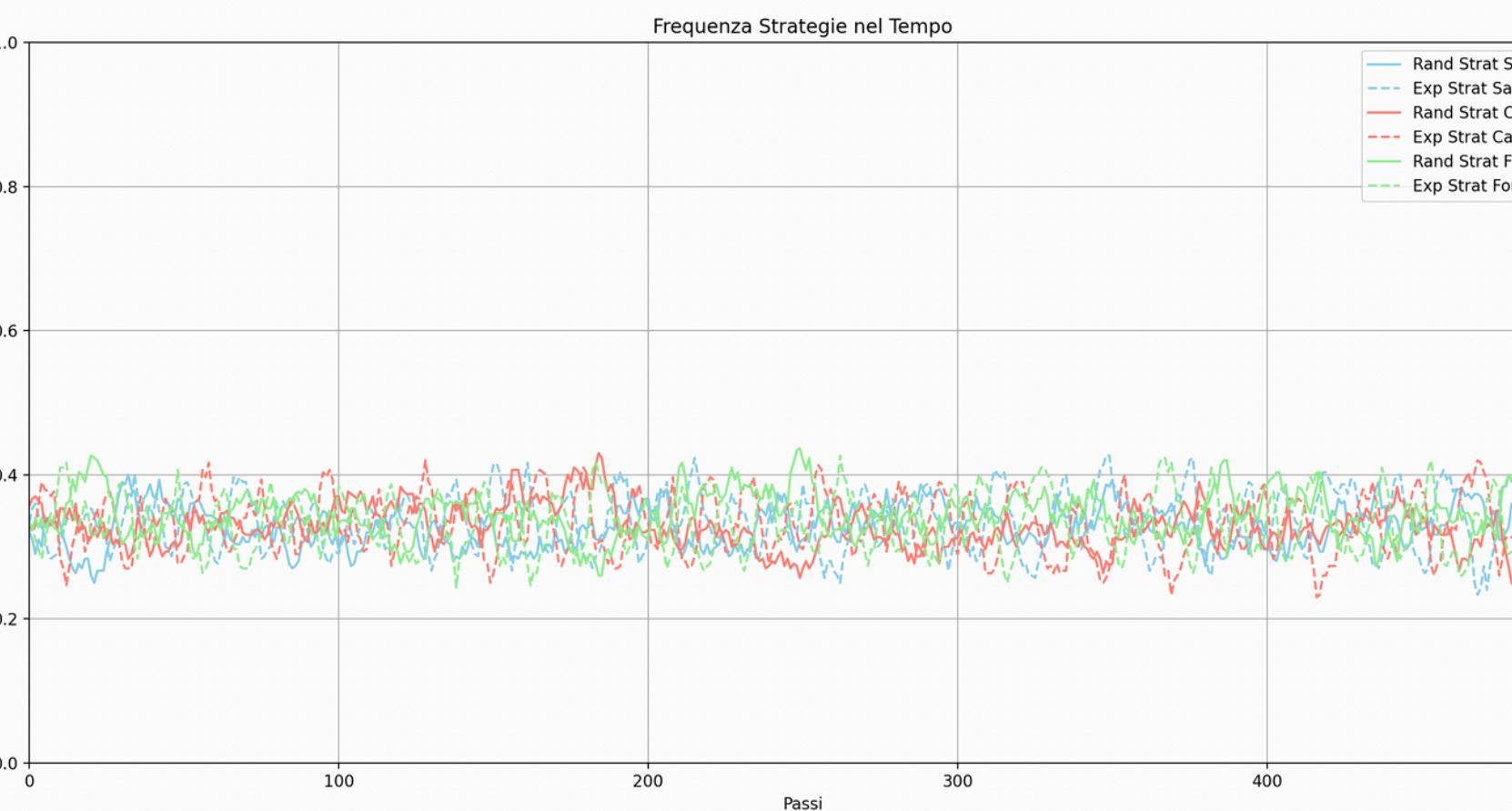
Evoluzione delle strategie nel tempo - Metodo di Guadagno: random



Evoluzione delle strategie nel tempo - Metodo di Guadagno: expected



Confronto Evoluzione Strategie  
Metodo Random vs Expected



## Statistiche Finali: Metodo Random vs Expected

### Metodo Random

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	0.114	0.635	105	0.35	1	-1
Carta	0.099	0.663	101	0.337	1	-1
Forbice	-0.064	0.698	94	0.313	1	-1
--- Totali / Medi ---						
Cambi strategia / step	29.416					
Tempo medio a equilibrio	Mai raggiunto					

### Metodo Expected

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	-0.051	0.001	90	0.3	0.06	-0.06
Carta	-0.001	0.0	112	0.373	0.06	-0.06
Forbice	0.048	0.001	98	0.327	0.06	-0.06
--- Totali / Medi ---						
Cambi strategia / step	32.06					
Tempo medio a equilibrio	Mai raggiunto					

# Osservazioni



## CONFRONTO

- L'introduzione del rumore fa **diminuire le frequenze**.
- Le **strategie competono maggiormente** ma nessuna strategia prevale.
- **Equilibrio misto**: Nessuna strategia domina. Le strategie convivono in frequenze stabili (es. tutte e tre ~0.33).
- **Viene a mancare l'andamento sinusoidale**.

# Simulazione 4: Influenza delle configurazioni iniziali delle strategie

Qual è l'effetto delle condizioni iniziali delle strategie sulla velocità di convergenza verso l'equilibrio?

**Parametri:**

**Numero di agenti:** 300

**Distribuzione iniziale equa:** [150, 100, 50] → DISTRIBUZIONE SBILANCIATA

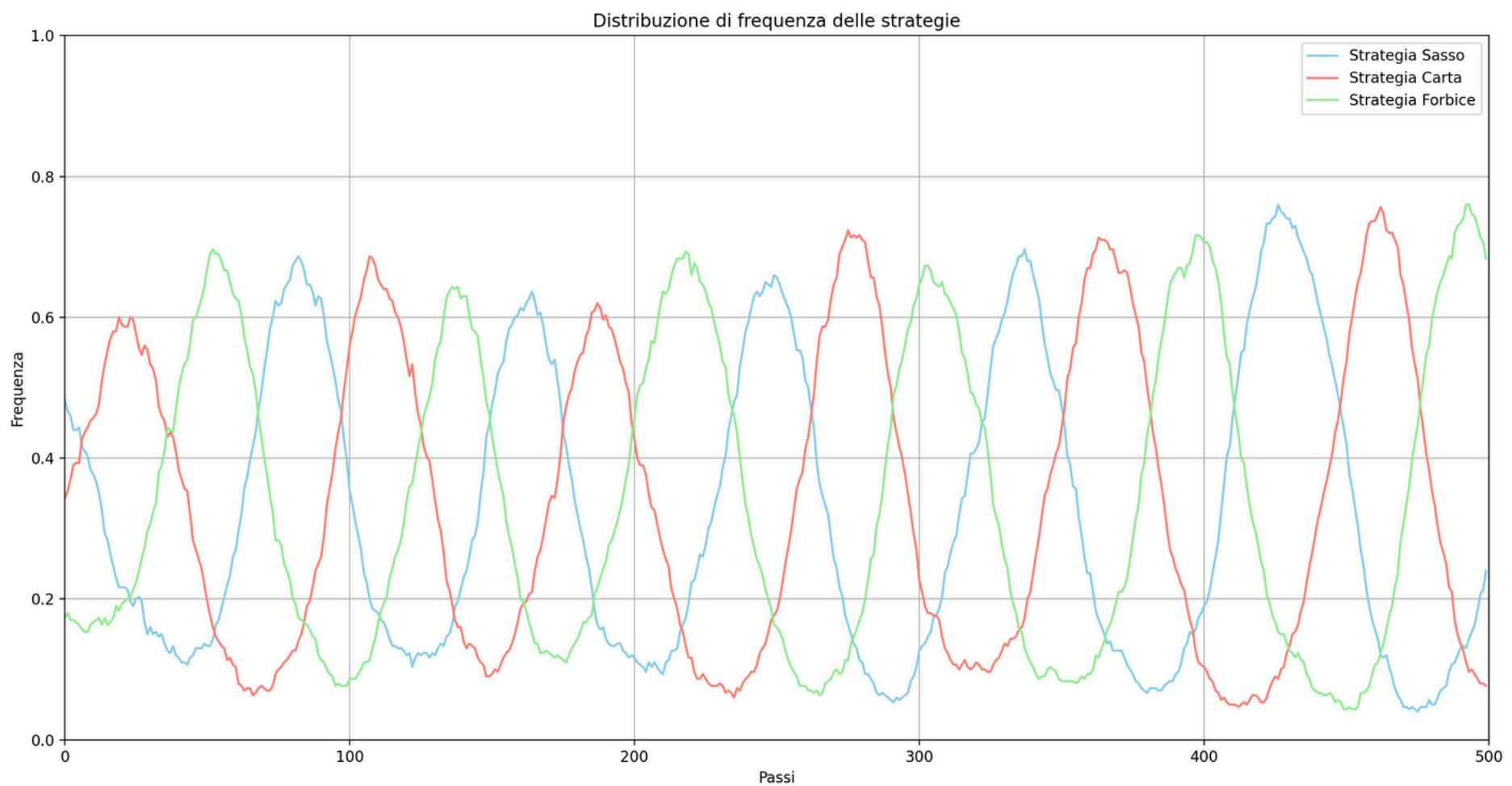
**Matrice dei payoff** = ([[0, -1, 1],  
[1, 0, -1],  
[-1, 1, 0]])

**Noise** = 0.01

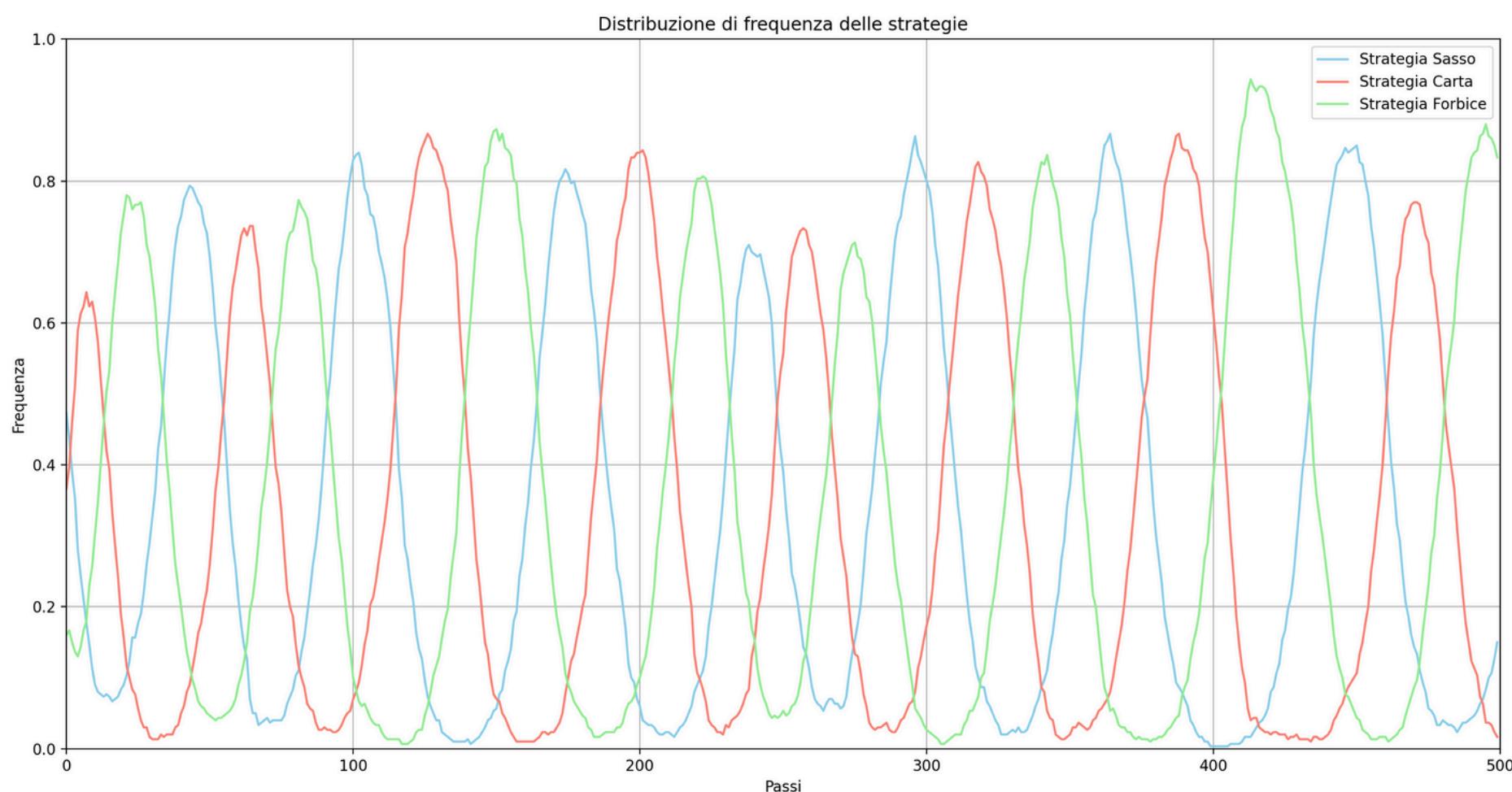
**Prob\_revision** = 0.2

**Steps** = 500

**Evoluzione delle strategie nel tempo - Metodo di Guadagno: random**



**Evoluzione delle strategie nel tempo - Metodo di Guadagno: expected**



## Statistiche Finali: Metodo Random vs Expected

### Metodo Random

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	-0.02	0.666	102	0.34	1	-1
Carta	0.023	0.772	88	0.293	1	-1
Forbice	-0.009	0.609	110	0.367	1	-1
--- Totali / Medi ---						
Cambi strategia / step	13.018					
Tempo medio a equilibrio	Mai raggiunto					

### Metodo Expected

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	0.05	0.026	242	0.807	0.083	-0.727
Carta	0.527	0.052	24	0.08	0.643	0.083
Forbice	-0.727	0.0	34	0.113	-0.727	-0.727
--- Totali / Medi ---						
Cambi strategia / step	13.014					
Tempo medio a equilibrio	Mai raggiunto					

# Osservazioni



- Le condizioni iniziali delle strategie **influenzano l'evoluzione**, ma non la velocità di convergenza verso l'equilibrio.
- In entrambi i metodi, la strategia opposta a Sasso, ovvero Carta, inizia a dominare ma poi si ritorna verso un andamento sinusoidale.
- L'evoluzione ha mostrato un **effetto di riequilibrio**: le strategie si sono distribuite in modo più omogeneo man mano che gli agenti hanno adattato le proprie scelte sulla base del payoff ricevuto.
- Vantaggio iniziale non garantisce dominanza finale se non accompagnato da un vantaggio nella matrice di payoff.
- **Stesso numero di passi.**

# Considerazioni

***Ci sono situazioni che modificano questo andamento?***

***In quale tipo di payoff-to-use prima?***

Nonostante i parametri variati (rumore, probabilità di revisione, distribuzione iniziale), tutte le simulazioni mostrano un **comportamento oscillatorio delle strategie**, caratterizzato da un **andamento sinusoidale persistente**. Questo riflette la **natura ciclica del gioco carta-forbice-sasso**, che porta a un **equilibrio dinamico** instabile piuttosto che a un punto fisso di equilibrio.

→ Il **rumore** pare essere l'unico elemento che porta ad un equilibrio misto in cui nessuna strategia domina. Le strategie convivono in frequenze stabili (es. tutte e tre ~0.33).

Anche se esiste un equilibrio teorico con frazioni costanti 1/3 per ciascuna strategia, questo non è evolutivamente stabile: basta una piccola perturbazione perché il sistema entri in una corsa ciclica infinita.

# Considerazioni

*E se invece volessi ottenere un equilibrio puro con una strategia dominante?*

# Simulazione 5: Raggiungimento di un equilibrio puro

Quali condizioni portano ad un equilibrio puro con matrice Sasso, carta, forbice?

**Parametri:**

**Numero di agenti:** 100

**Distribuzione iniziale equa:** [33, 33, 34]

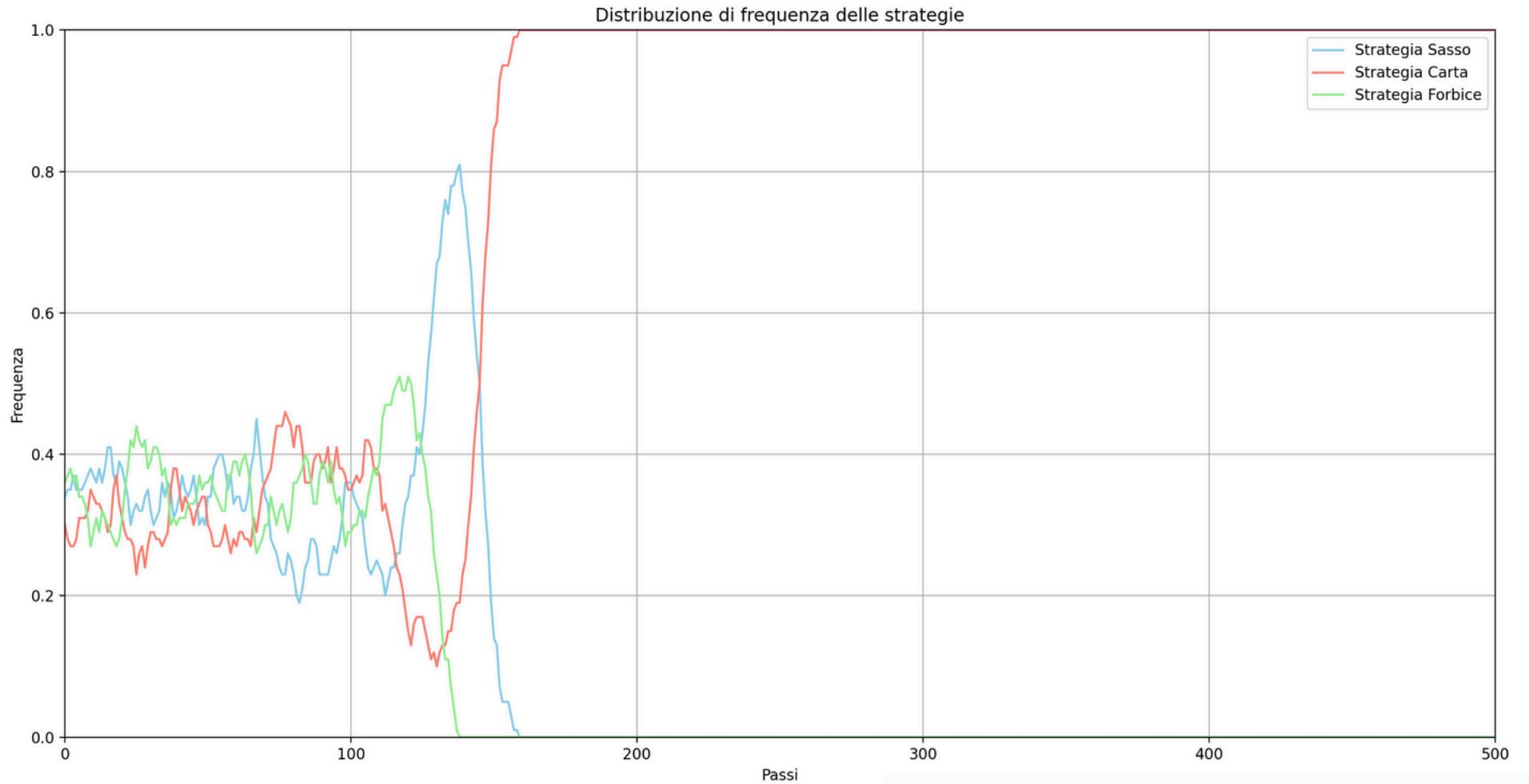
**Matrice dei payoff** = ([[0, -1, 1],  
[1, 0, -1],  
[-1, 1, 0]])

**Noise** = 0

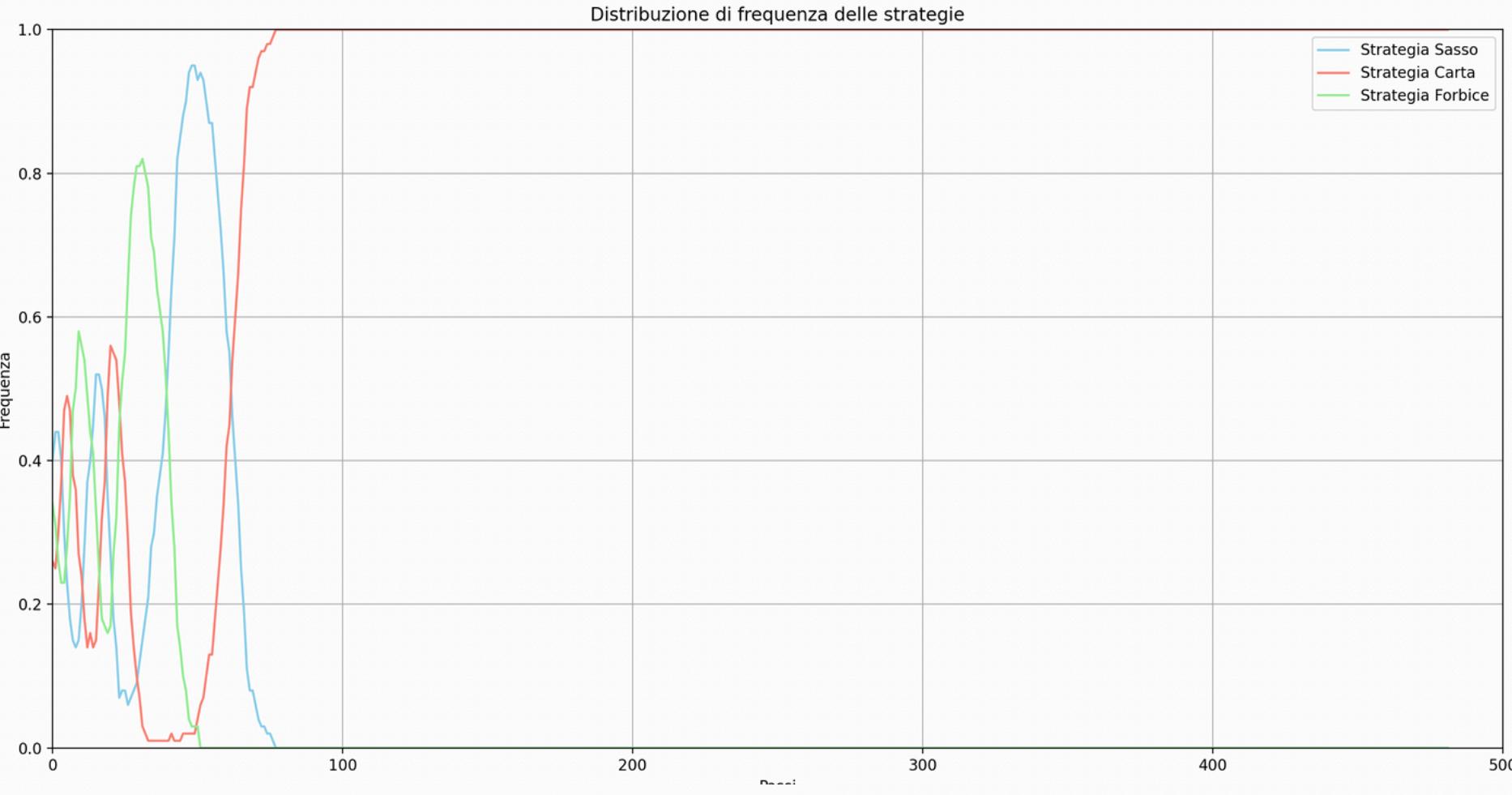
**Prob\_revision** = 0.3

**Steps** = 500

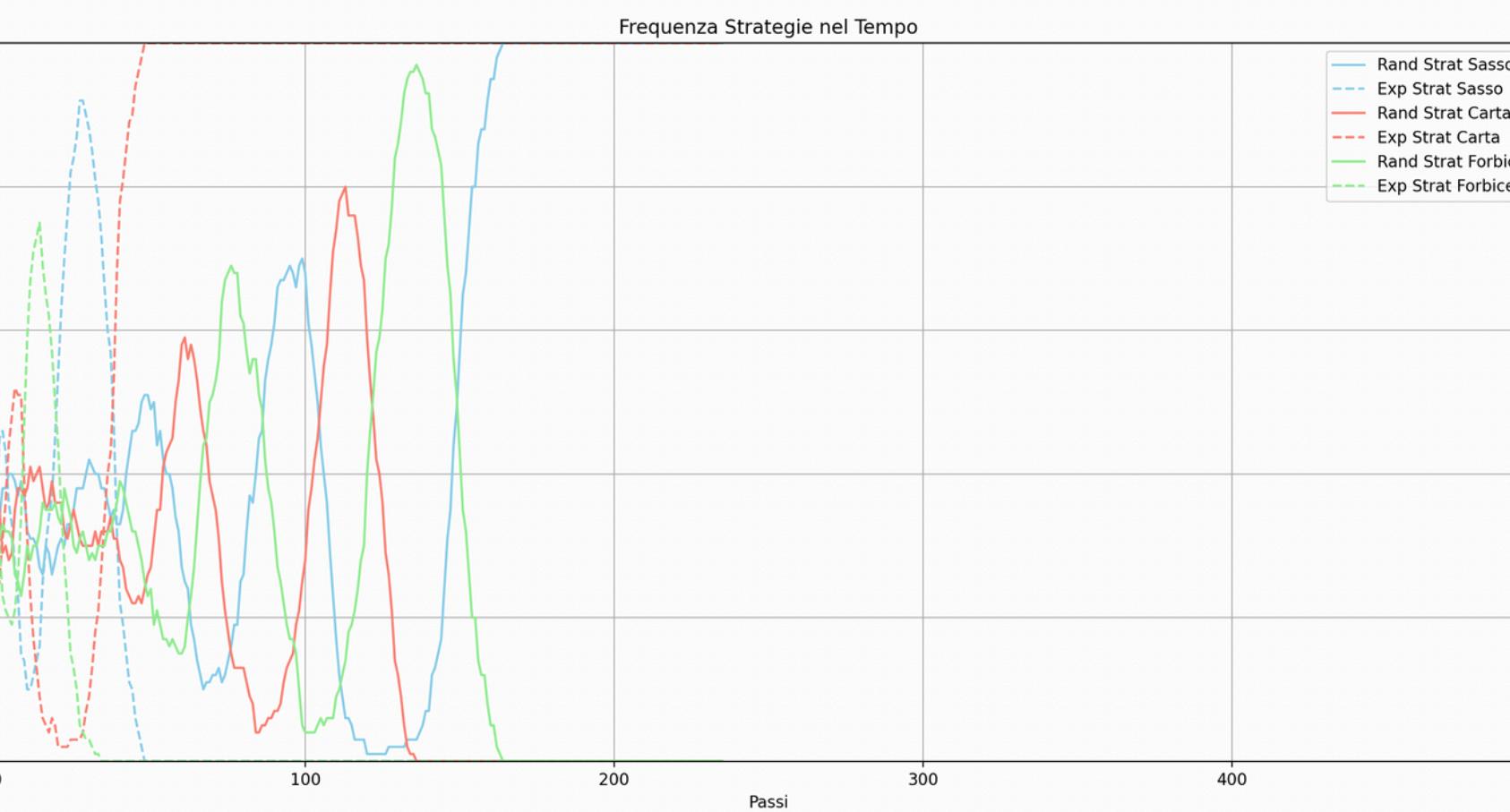
Evoluzione delle strategie nel tempo - Metodo di Guadagno: random



Evoluzione delle strategie nel tempo - Metodo di Guadagno: expected



Confronto Evoluzione Strategie  
Metodo Random vs Expected



## Statistiche Finali: Metodo Random vs Expected

### Metodo Random

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	0.0	0.0	100	1.0	0	0
Carta	0.0	0.0	0	0.0	0	0
Forbice	0.0	0.0	0	0.0	0	0
--- Totali / Medi ---						
Cambi strategia / step	3.4427480916030535					
tempo medio a equilibrio	136					

### Metodo Expected

Strategia	Media Payoff	Varianza Payoff	Numero di Giocatori	Frequenza	Payoff Massimo	Payoff Minimo
Sasso	0.0	0.0	0	0.0	0.0	0.0
Carta	0.0	0.0	100	1.0	0.0	0.0
Forbice	0.0	0.0	0	0.0	0.0	0.0
--- Totali / Medi ---						
Cambi strategia / step	1.2251908396946565					
tempo medio a equilibrio	50					

# Considerazioni

*Quali valori dei parametri permettono di ottenere  
un equilibrio puro con una strategia dominante?*

- **Eliminando il rumore e diminuendo il numero di agenti**, dopo un numero relativamente basso di step, si ottiene un equilibrio puro.
- **Nel metodo expected si raggiunge l'equilibrio nella metà del tempo rispetto al metodo random.**



# Grazie per l'attenzione!

---

---