

# Ethical Simulation in Autonomous Driving

## ***Introduction:***

This study endeavors to develop an ethical simulator for an Artificial Intelligence (AI) system deployed in autonomous vehicles. The primary aim is to investigate the behavior of the AI in relation to the renowned ethical dilemma known as the "trolley problem." The simulation has been meticulously designed to analyze the decision-making processes of the AI based on three pivotal variables: ethical value, the age of the individuals involved, and the AI's preference between younger and older individuals.

## ***Methodology:***

We have developed a C++ code representing the AI of an autonomous vehicle, which models the trolley problem through two groups of individuals, A and B. The AI must decide which group to sacrifice, utilizing evaluation algorithms that consider the ethical weight of the machine and variables such as the ages of the individuals involved. To ensure ethical neutrality, it analyzes both options without preconceived biases.

This AI is a rule-based ethical decision-making algorithm that computes scores based on age and ethical weight, facilitating an unbiased selection between the groups. It is designed to explore insights into the behavior of AI in complex moral situations through iterative simulations.

## ***Study Variables:***

The primary variables considered in the simulation include the ethical value, quantified on a scale from 0 to 10, the ages of individuals within groups A and B, and the AI's preferential inclination towards sacrificing younger or older individuals. These variables are integral to comprehending how the AI formulates ethical decisions in complex scenarios.

## ***Preliminary Results:***

Through 10,000 iterations of the simulation, we examined the behavior of the AI in response to diverse ethical scenarios. The outcomes, represented in terms of scores and the choices made by the machine, were systematically analyzed to elucidate the influence of the identified variables on ethical decision-making. To ensure statistically robust data, we subsequently conducted an extended analysis involving 1,170,000 iterations, further enhancing the reliability and validity of our findings.

```
#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>
```

*/\* Questo codice simula un'IA di una macchina a guida autonoma (Intelligenza Artificiale) che deve prendere una decisione etica*

*basandosi su un algoritmo di valutazione. Il problema del trolley è modellato attraverso due gruppi di persone, A e B, e l'IA deve decidere quale gruppo sacrificare per massimizzare il proprio punteggio etico. L'algoritmo tiene conto del peso etico della macchina e di variabili come l'età delle persone nei gruppi. L'obiettivo è esplorare il comportamento dell'IA in differenti scenari, considerando le sue scelte e i risultati ottenuti. Al fine di ottenere risultati oggettivi non sono stati autoimposti principi etici particolari come quelli umani (che hanno come priorità salvare i più giovani), bensì la macchina ha la possibilità di analizzare anche il caso opposto, ottenendo così risultati prettamente matematici senza alcun pregiudizio.*

*\*/*

***// Definizione della struttura Macchina***

```
struct Macchina {
    int punteggio;
    float PesoEtico;
    float MoltEta;
    float risultatoA = 0;
    float risultatoB = 0;
    bool Priori;
    std::string scelta;
};
```

***// Funzione per calcolare il punteggio di un individuo macchina per il gruppo A***

```
float CalcolaRisultatoA(const Macchina& macchina, const
std::vector<int>& gruppoA, bool priori) {
    float risultatoA = 0;
    for (size_t i = 0; i < gruppoA.size(); ++i) {
        // Calcolo del valore di MoltEta per gruppo A
        float moltEtaA = priori ? (5.0f - (gruppoA[i] / 100.0f) *
4.0f) : (1.0f + (gruppoA[i] / 100.0f) * 4.0f);
        risultatoA += gruppoA[i] * macchina.PesoEtico * moltEtaA;
    }
    return risultatoA;
}
```

***// Funzione per calcolare il punteggio di un individuo macchina per il gruppo B***

```
float CalcolaRisultatoB(const Macchina& macchina, const
std::vector<int>& gruppoB, bool priori) {
    float risultatoB = 0;
```

```

        for (size_t i = 0; i < gruppoB.size(); ++i) {
            // Calcolo del valore di MoltEta per gruppo B
            float moltEtaB = priori ? (5.0f - (gruppoB[i] / 100.0f) *
4.0f) : (1.0f + (gruppoB[i] / 100.0f) * 4.0f);
            risultatoB += gruppoB[i] * macchina.PesoEtico * moltEtaB;
        }
        return risultatoB;
    }

int main() {
    // Inizializzazione del generatore di numeri casuali
    std::srand(static_cast<unsigned>(std::time(0)));
    std::string uccisione;

    // Creazione di una popolazione di 10 macchine
    const int numMacchine = 10;
    std::vector<Macchina> popolazione(numMacchine);

    for (int iterazione = 1; iterazione <= 10000; ++iterazione) {

        // Riempi vettore popolazione con valori casuali
        for (int i = 0; i < numMacchine; ++i) {
            popolazione[i].punteggio = 0;
            popolazione[i].PesoEtico = (static_cast<float>(std::rand()
%
            5) + 1); // Valore tra 0 e 10
            popolazione[i].MoltEta = 0; // Inizializzato a 0
            popolazione[i].Priori = (std::rand() % 2 == 0); // Scegli
casualmente vero o falso
        }

        // Creazione e riempimento del vettore A
        const int grandezzaA = std::rand() % 10 + 1;
        std::vector<int> A(grandezzaA);
        for (int i = 0; i < grandezzaA; ++i) {
            A[i] = std::rand() % 76 + 5; // Valore tra 5 e 80
        }

        // Creazione e riempimento del vettore B
        const int grandezzaB = std::rand() % 10 + 1;
        std::vector<int> B(grandezzaB);
        for (int i = 0; i < grandezzaB; ++i) {
            B[i] = std::rand() % 76 + 5; // Valore tra 5 e 80
        }

        // Calcola il punteggio per ogni individuo macchina e stampa i risultati
        for (auto& macchina : popolazione) {
            macchina.risultatoA = CalcolaRisultatoA(macchina, A,
macchina.Priori);

```

```
        macchina.risultatoB = CalcolaRisultatoB(macchina, B,
macchina.Priori);
```

```
// Confronta i risultati e aggiorna il punteggio e la scelta
```

```
    if (macchina.risultatoA > macchina.risultatoB)
    {

        if(A.size() > B.size()){
            uccisione = "0 ";
            macchina.punteggio += A.size();
        }
        if(A.size() < B.size()){
            uccisione = "1 ";
            macchina.punteggio += B.size();
        }

        macchina.scelta = "Uccide il gruppo B ";
    } else if (macchina.risultatoB > macchina.risultatoA) {
        if(A.size() < B.size()){
            uccisione = "0 ";
            macchina.punteggio += B.size();
        }
        if(A.size() > B.size()){
            uccisione = "1 ";
            macchina.punteggio += A.size();
        }

        macchina.scelta = "Uccide il gruppo A ";
    } else {

        macchina.scelta = "Nessuna scelta ";
    }

    if(A.size() == B.size()){
        uccisione = " ";
        macchina.punteggio += A.size();
    }

    // Stampa i risultati
    std::cout << uccisione << std::endl;
}

return 0;

}
```

Data Following 1,170,000 Iterations:

Iterazioni totali	Risultati: 0	% 0	Risultati: 1	% 1
90.000,00	82.548,00	91,72%	7.452,00	8,28%
90.000,00	82.998,00	92,22%	7.002,00	7,78%
90.000,00	82.665,00	91,85%	7.335,00	8,15%
90.000,00	82.926,00	92,14%	7.074,00	7,86%
90.000,00	82.683,00	91,87%	7.317,00	8,13%
90.000,00	82.620,00	91,80%	7.380,00	8,20%
90.000,00	82.809,00	92,01%	7.191,00	7,99%
90.000,00	82.755,00	91,95%	7.245,00	8,05%
90.000,00	82.674,00	91,86%	7.326,00	8,14%
90.000,00	82.809,00	92,01%	7.191,00	7,99%
90.000,00	82.782,00	91,98%	7.218,00	8,02%
90.000,00	82.476,00	91,64%	7.524,00	8,36%
90.000,00	82.332,00	91,48%	7.668,00	8,52%
1.170.000,00	1.075.077,00	91,89%	94.923,00	8,11%

Dev st.	Dev st.
180,43	180,43
0,22%	2,47%

0=Cambiamento del percorso

1=Mantenimento del percorso

## Conclusions:

A crucial aspect of our study was the comparison between the ethical decisions made by the AI and the human preferences gathered through a preliminary questionnaire. Before implementing the simulator, we interviewed 150 individuals, asking them what choice they would make in similar situations, varying the number of people involved. Surprisingly, 93% of the participants indicated that they would choose to change the trolley's course when a larger number of individuals were at stake.

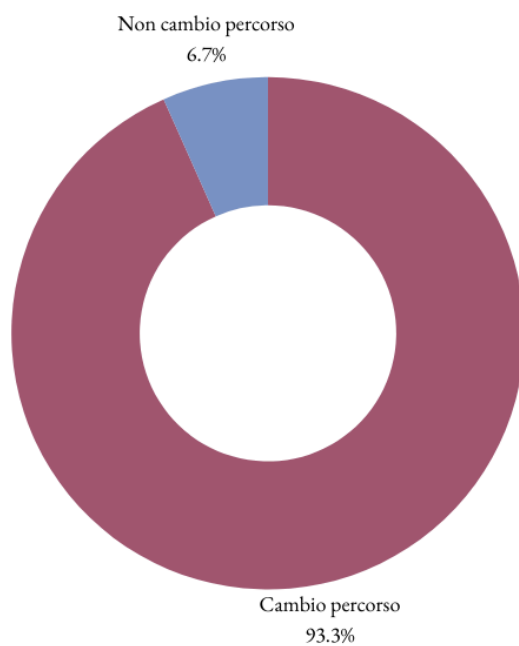
Even more significant is that during our simulations, the machine opted to change the trolley's course at the same rate of 93%. This alignment between human preferences and the AI's decisions underscores the effectiveness of our simulator in capturing the ethical inclinations of individuals.

The importance of this alignment between the simulation and human preferences lies in the fact that we have obtained a model that faithfully reflects the choice variables of people. This result suggests

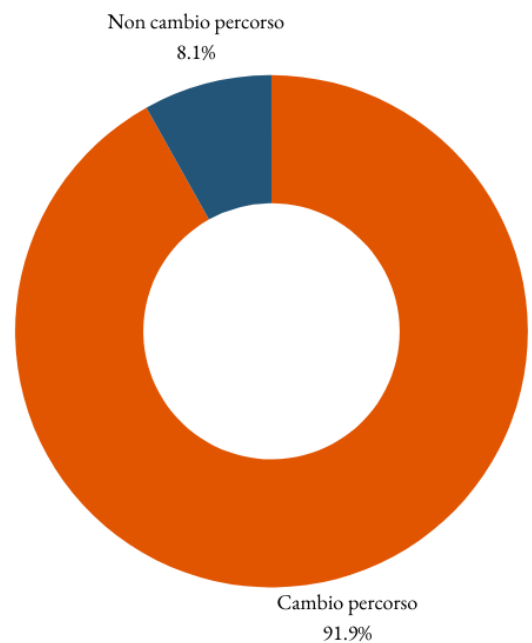
that our simulator is capable of successfully replicating human decision-making dynamics in complex ethical contexts, offering valuable insights into how AI can reflect and respond to societal preferences in critical situations.

In conclusion, our simulation not only provides a detailed framework of the ethical decisions made by the AI but also demonstrates its ability to adequately replicate the will and choice variables of individuals. This advancement is crucial for developing autonomous systems that align with societal preferences, thereby contributing to ethically responsible and socially accepted autonomous driving.

Problema del trolley: Risposte umane  
Totale: 150



Problema del trolley: Risposte AI  
Totale: 1.170.000



-Andrea G. Gitto  
-Simone Piconese