

---

# **Entrenamiento evolutivo de agentes pastores para control de rebaños en un entorno multiagente**

---

**G. Cagnola, F. G. Hergenreder y L. Saurin**

Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral  
Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional, UNL-CONICET

## **Resumen**

En los sistemas de pastoreo tradicionales, los rebaños de ovejas se desplazan libremente por el terreno mientras consumen vegetación, y el perro pastor actúa como guía y protector, manteniéndolas agrupadas y dirigiendo su movimiento hacia zonas específicas. Tomando inspiración en este escenario real, el presente proyecto busca entrenar agentes artificiales capaces de desempeñar un rol similar, con el objetivo de estudiar estrategias que puedan ser útiles para la gestión coordinada de grupos, por ejemplo en situaciones de evacuación.

El movimiento de las ovejas se modela mediante un enfoque heurístico, que reproduce el comportamiento grupal a partir de reglas simples de interacción local, mientras que el agente pastor es controlado por una red neuronal multicapa cuya arquitectura se basa en trabajos previos. Los pesos de esta red se optimizan mediante un algoritmo genético, permitiendo que el comportamiento del agente surja de manera emergente sin imponer reglas explícitas de control.

## **1. Introducción**

Dentro del campo de la inteligencia artificial, las redes neuronales artificiales, y en particular los perceptrones multicapa (MLP), se encuentran entre los modelos más utilizados para aproximar funciones complejas o aprender políticas de control a partir de datos. En este trabajo se utiliza un MLP como controlador del agente pastor dentro de un entorno de simulación multiagente de pastoreo.

A diferencia de los enfoques deterministas basados en reglas, aquí el comportamiento del pastor no se programa explícitamente. En su lugar, se adopta un enfoque de optimización evolutiva, donde un Algoritmo Genético (GA) se encarga de optimizar los pesos de la red neuronal para maximizar una función de aptitud que refleja la calidad del pastoreo.

El objetivo principal de este proyecto es evaluar si un pastor controlado por una red neuronal simple, con pesos optimizados mediante un Algoritmo Genético, puede realizar estrategias efectivas de agrupamiento, conducción y aproximación al objetivo, interactuando con un rebaño que sigue un modelo heurístico de comportamiento colectivo.

## **2. Métodos**

El flujo de datos del proyecto se divide en dos fases principales, cada una con un objetivo distinto: el pipeline de entrenamiento (offline), donde se aprende el comportamiento, y el pipeline de ejecución (online), donde se aplica en la simulación el comportamiento aprendido.

### **2.1. Pipeline de entrenamiento: algoritmo genético**

El objetivo de esta fase es encontrar el conjunto óptimo de pesos (genotipo) para la red neuronal del pastor mediante el proceso evolutivo del algoritmo genético.

Inicialización → Decodificación → Evaluación → Evolución → Repetir

**Representación genotípica:** cada individuo se representa por un genoma binario que codifica los pesos de la red neuronal. La decodificación transforma segmentos de bits en valores reales dentro del intervalo  $[-15, 16]$ .

**Evaluación:** para cada individuo se decodifican los pesos, se inicializa la red neuronal y se ejecuta una simulación completa del entorno. La evaluación se repite 2 veces con reubicaciones aleatorias para promediar la estocasticidad, pero sin afectar significativamente el tiempo de entrenamiento.

**Selección y operadores genéticos:** se aplica selección por ventana con elitismo; la descendencia se genera mediante cruce simple y mutación *bit-flip* con probabilidad del 5 %. El mejor individuo global se memoriza en cada generación.

**Criterio de parada:** pasos máximos por simulación o ciclos de reloj sin modificar posición.

## 2.2. Pipeline de ejecución: red neuronal

El objetivo de esta fase es aplicar en tiempo real el comportamiento aprendido por el mejor pastor durante la simulación visual, decidiendo el movimiento en cada paso de tiempo. El objetivo es que el pastor logre reunir todas las ovejas al *target* mediante recolección y guiado de las ovejas.

Inicialización → Preprocesamiento → Propagación → Salida → Postprocesamiento

**Carga del modelo:** se carga el genoma del individuo y se decodifican sus pesos para inicializar la red ShepherdNN.

**Percepción:** en cada paso el pastor recibe información sobre su entorno (objetivo y ovejas) para guiarlas.

**Preprocesamiento:** todas las coordenadas se expresan en forma relativa al pastor y se normalizan por la longitud de la diagonal del mundo.

**Red neuronal:** se utiliza un MLP con propagación hacia adelante para obtener la dirección en la cual se debe mover (ver Sección 2.3.2).

## 2.3. Entorno y comportamiento

El sistema de pastoreo se implementa en un entorno de simulación multiagente bidimensional que define la dinámica física y social de los agentes. El mundo se define como un espacio rectangular de dimensiones configurables, donde todos los agentes operan sin límites físicos de rebote.

### 2.3.1. Comportamiento del rebaño

El movimiento de las ovejas se basa en el modelo heurístico de Strömbom et al. [1]. Este modelo aplica tres fuerzas vectoriales para lograr el comportamiento colectivo y evasivo:

- **Repulsión local:** fuerza de repulsión entre ovejas, activada si la distancia es menor al radio de repulsión. Esto evita la superposición y mantiene una densidad baja.
- **Repulsión del pastor:** fuerza de huida generada cuando un pastor se acerca a una oveja dentro del radio de repulsión del pastor. Esta fuerza activa el estado de ser “pastoreada”.
- **Cohesión al centro de gravedad:** fuerza de atracción hacia el centro de gravedad de los 7 vecinos más cercanos. Esta fuerza solo se activa cuando la oveja está siendo pastoreada.
- **Ruido:** el movimiento se complementa con un factor de inercia y un componente de ruido angular para introducir un comportamiento estocástico.

$$\mathbf{v}_i^{t+1} = h \mathbf{v}_i^t + \sum_{j \in N_i} \mathbf{f}_{ij}^{(aa)} + \mathbf{f}_i^{(s)} + \alpha (\mathbf{c}_i - \mathbf{s}_i) + \eta \quad (1)$$

$$\mathbf{s}_i^{t+1} = \mathbf{s}_i^t + \Delta t \mathbf{v}_i^{t+1} \quad (2)$$

### 2.3.2. Comportamiento del pastor

El pastor actúa como el agente de control cuyo comportamiento es emergente y se optimiza mediante el algoritmo genético.

**Arquitectura:** la arquitectura utilizada se basa en las redes neuronales empleadas por Napolitano et al. [2]. Esta consta de 256 neuronas en la capa de entrada y 128 neuronas en la capa de salida, ambas con función de activación no lineal ReLU.

**Entradas:** El pastor procesa un vector de estado normalizado que incluye la posición del target, las posiciones de las 5 ovejas más cercanas, la posición del centroide de ovejas. Todas relativas a sí mismo y normalizadas por la diagonal.

**Salidas:** la salida de la red neuronal se convierte en un vector dirección  $dx-dy$  y se multiplica por una distancia de paso constante para actualizar su posición.

## 2.4. Función de aptitud y criterios de rendimiento

Para cada individuo  $i$  con genoma  $g_i$ , su aptitud se define como la suma ponderada de cinco términos evaluados a lo largo de  $R$  repeticiones independientes en la simulación:

$$F(g_i) = w_c \bar{C} + w_g \bar{G} + w_i \bar{I} + w_d \bar{D} + w_f \bar{F}, \quad (3)$$

donde cada término con barra superior representa el promedio sobre las simulaciones:

$$\bar{X} = \frac{1}{R} \sum_{r=1}^R X_r \quad (4)$$

Los componentes individuales son:

**Cohesión ( $C$ ).** Mide qué tan cerca están las ovejas entre sí:

$$C_r = \frac{1}{N} \sum_{n=1}^N d(s_n, c) \quad (5)$$

donde  $N$  es la cantidad de ovejas,  $d(\cdot)$  la distancia euclíadiana,  $s_n$  la posición de la oveja  $n$  y  $c$  el centroide del rebaño.

**Progreso hacia el objetivo ( $G$ ).** Mide cuánto acercó el centroide de las ovejas al objetivo:

$$G_r = \max \left( 0, \frac{d_i - d_f}{d_i} \right) \quad (6)$$

donde  $d_i$  es la distancia inicial y  $d_f$  la distancia final al objetivo.

**Ovejas dentro del objetivo ( $I$ ).** Mide qué proporción de ovejas termina dentro del radio:

$$I_r = \frac{1}{N} \sum_{n=1}^N \mathbf{1}[d(s_n, o) \leq R_{\text{obj}}] \quad (7)$$

donde  $o$  es la posición del centro del objetivo,  $R_{\text{obj}}$  su radio y  $\mathbf{1}[\cdot]$  la función indicadora que contabiliza las ovejas dentro del objetivo.

**Conducción ( $D$ ).** Mide qué tan bien el pastor empuja las ovejas hacia el objetivo, inspirada en la fase de *driving* del modelo de Strömbom [1]. Se utiliza una métrica normalizada provista por el simulador, con  $D_r \in [0, 1]$ .

**Finalización ( $F$ ).** Bonus si se completa la tarea:

$$F_r = \begin{cases} 1, & \text{si todas las ovejas llegan al objetivo,} \\ 0, & \text{en caso contrario.} \end{cases} \quad (8)$$

### 3. Resultados

#### 3.1. Desempeño del entrenamiento

Se presenta un gráfico del mejor *fitness* obtenido a lo largo de 2000 y 100 generaciones.

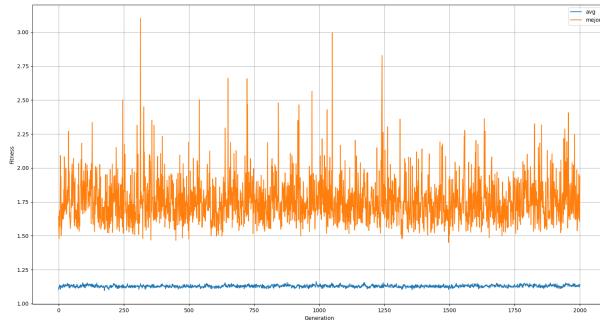


Figura 1: Evolución del fitness a lo largo de 2000 generaciones.

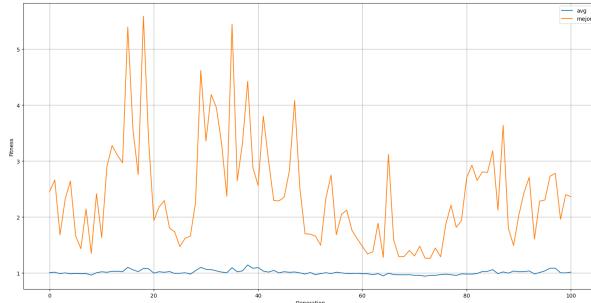


Figura 2: Evolución del fitness a lo largo de 100 generaciones.

Se muestran dos curvas correspondientes al mejor fitness y al fitness promedio durante un entrenamiento de 500 generaciones (arriba) y 100 generaciones (abajo). Como se puede ver, estos siguen un comportamiento errático, con picos y valles y sin una tendencia clara. La aptitud promedio se mantiene alrededor del valor 1, lo cual indica que la gran mayoría de los genomas generados por el AG resultan en pastores que fallan en la tarea de pastoreo. Es probable que estos agentes se vayan por fuera del área definida o no logren mover el centro de gravedad del rebaño de manera efectiva, lo que los penaliza fuertemente en términos de progreso y finalización.

Respecto de la aptitud del mejor individuo, la característica más notable de esta curva es su extrema volatilidad. Los picos de aptitud alcanzan consistentemente valores altos (superiores a 5), lo que demuestra que la función de aptitud multifactorial puede lograr altas puntuaciones, es decir, el potencial para el buen comportamiento existe: el agente fue cohesivo, progresó y condujo bien; sin embargo las caídas inmediatas desde los picos hasta la línea base de 1 en la siguiente generación indican que es altamente susceptible a la posición aleatoria inicial de las ovejas y el pastor.

### 3.2. Métricas de rendimiento

En la siguiente tabla se presentan los valores obtenidos del fitness y valores individuales de sus componentes, en 10 generaciones de las 600 con las que se entrenó. Estas generaciones seleccionadas corresponden a los casos en los que se obtuvo una mejora del fitness.

Para analizar la evolución del comportamiento aprendido por el pastor, se compararon los valores de cada uno de los componentes del fitness por separado, para ver su progreso individual.

Generación	Fitness	Cohesión	To_goal	Inside	Driving	Finish
1	1.28018	0.85522	0.00013	0.12483	0.30000	0.00000
4	1.31967	0.84516	0.00006	0.14333	0.33111	0.00000
11	1.46010	0.88023	0.00026	0.00350	0.57611	0.00000
12	1.55040	0.86507	0.00022	0.07067	0.61444	0.00000
23	1.67683	0.86841	0.00037	0.03583	0.77222	0.00000
29	1.74726	0.87782	0.00049	0.04950	0.81944	0.00000
34	1.77110	0.90313	0.00041	0.08200	0.78556	0.00000
206	1.79259	0.89134	0.00070	0.03833	0.86222	0.00000
329	1.79317	0.88970	0.00063	0.02117	0.88167	0.00000
337	1.79577	0.87917	0.00054	0.04050	0.87556	0.00000

Cuadro 1: Evolución de los componentes del fitness en generaciones con mejora.

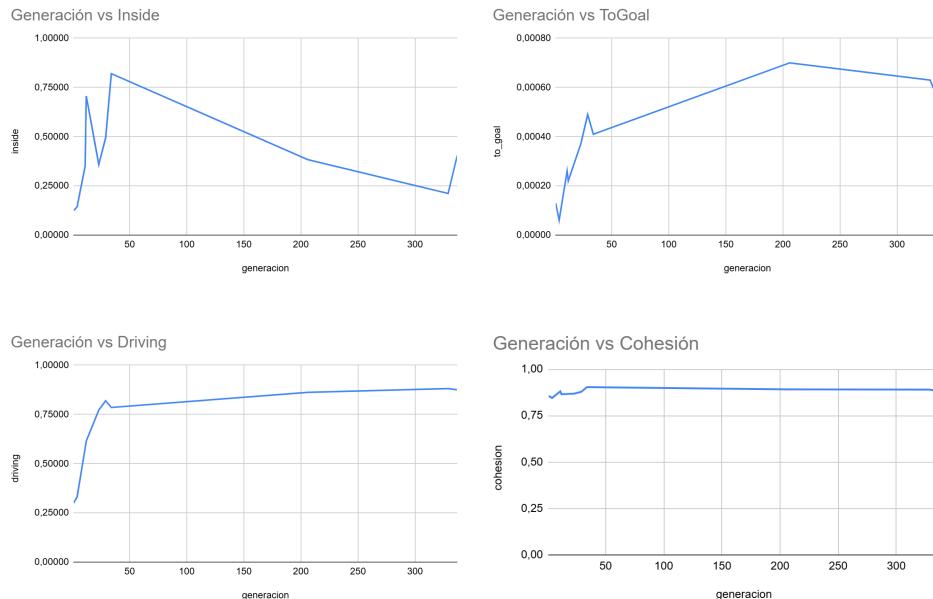


Figura 3: Evolución de los componentes del fitness a lo largo del entrenamiento.

La cohesión se mantuvo en un promedio de 0,87 y con una recta constante prácticamente desde el inicio, lo cual indica que el pastor logró mantener juntas a las ovejas.

El progreso muestra valores muy bajos (0,00006 a 0,00070), lo cual indica que si bien el pastor conduce activamente, el centroide no avanza significativamente hacia el objetivo en el tiempo disponible.

El componente Driving es el que muestra el incremento más constante, pasando de 0,3 en la generación 1 a 0,88 en la generación 329.

La métrica Inside se mantiene baja y errática, y Finish permanece en 0, evidenciando que el rebaño no llega a ingresar completamente en el objetivo dentro del tiempo simulado.

### 3.3. Comportamiento emergente del pastor

Durante las simulaciones se observaron patrones de comportamiento que no fueron programados explícitamente, sino que emergieron como resultado de la combinación y el ajuste evolutivo de los términos del fitness. A medida que avanzó el entrenamiento, el agente pastor desarrolló estrategias efectivas para reunir al rebaño, más no para el guiado del mismo hacia el objetivo.

En las primeras generaciones, el agente tendía a moverse de manera errática o desplazarse fuera del área de interés, pero a lo largo de la evolución logró moverse de forma de agrupar las ovejas.

No pudieron observarse comportamientos en los que el pastor, además de agrupar el rebaño, lo guíe efectivamente al objetivo.

### 3.4. Robustez y generalización

Para evaluar la robustez del modelo y su capacidad de generalización, se realizaron ensayos adicionales variando distintos parámetros del entrenamiento evolutivo. En particular, se modificaron el tamaño de la población, la cantidad de generaciones, la cantidad de progenitores y el número de evaluaciones redundantes.

En todos los casos no se observaron mejoras significativas respecto del desempeño logrado en el entrenamiento principal. Incluso al incrementar sustancialmente los recursos evolutivos, por ejemplo, ejecutando un entrenamiento extendido con 2000 generaciones, una población de 400 individuos y 9 ejecuciones por genoma (con posterior promedio para reducir el ruido estocástico), el comportamiento resultante no mostró una tendencia clara de convergencia hacia políticas superiores. Esto sugiere que el modelo actual no logra beneficiarse de un mayor esfuerzo computacional y que la estructura del espacio de soluciones, junto con la arquitectura de la red neuronal, podría estar limitando la capacidad del AG para descubrir estrategias de pastoreo más efectivas

### 3.5. Limitaciones del modelo

Aunque el sistema logró desarrollar ciertos patrones de comportamiento emergente, el modelo presenta una serie de limitaciones inherentes tanto al diseño del entorno como a la metodología de entrenamiento. Estas limitaciones son importantes para contextualizar el alcance del trabajo y proponer mejoras futuras, entre las cuales se encuentran:

- **Percepción limitada:** el pastor solo percibe las 5 ovejas más cercanas.
- **Dependencia del modelo heurístico:** aprende basado en Strömbom, lo que limita generalización.
- **Función de aptitud incompleta:** no penaliza trayectorias erráticas ni tiempos excesivos.
- **Costo computacional elevado:** el AG requiere evaluar miles de individuos y simulaciones repetidas.

## Conclusión

En este trabajo se desarrolló un sistema de pastoreo autónomo basado en un modelo heurístico para las ovejas y un pastor controlado por una red neuronal optimizada mediante un Algoritmo Genético. Los resultados muestran que el agente puede aprender comportamientos emergentes funcionales, aunque sin completar la tarea final dentro del tiempo simulado. El enfoque evolutivo se muestra válido y extensible a mejoras futuras.

## Referencias

- [1] Daniel Strömbom, Richard P. Mann, Alan M. Wilson, Stephen Hailes, A. Jennifer Morton, David J. T. Sumpter, and Andrew J. King. Solving the shepherding problem: heuristics for herding autonomous, interacting agents. *Journal of The Royal Society Interface*, 11(100):20140719, 2014. doi: 10.1098/rsif.2014.0719. URL <https://royalsocietypublishing.org/doi/epdf/10.1098/rsif.2014.0719>.

- [2] Italo Napolitano, Andrea Lama, Francesco De Lellis, and Mario Di Bernardo. Emergent cooperative strategies for multi-agent shepherding via reinforcement learning. In *2025 European Control Conference (ECC)*, pages 1809–1814, 2025. doi: 10.23919/ECC65951.2025.11186874. URL <https://arxiv.org/html/2411.05454v1>.