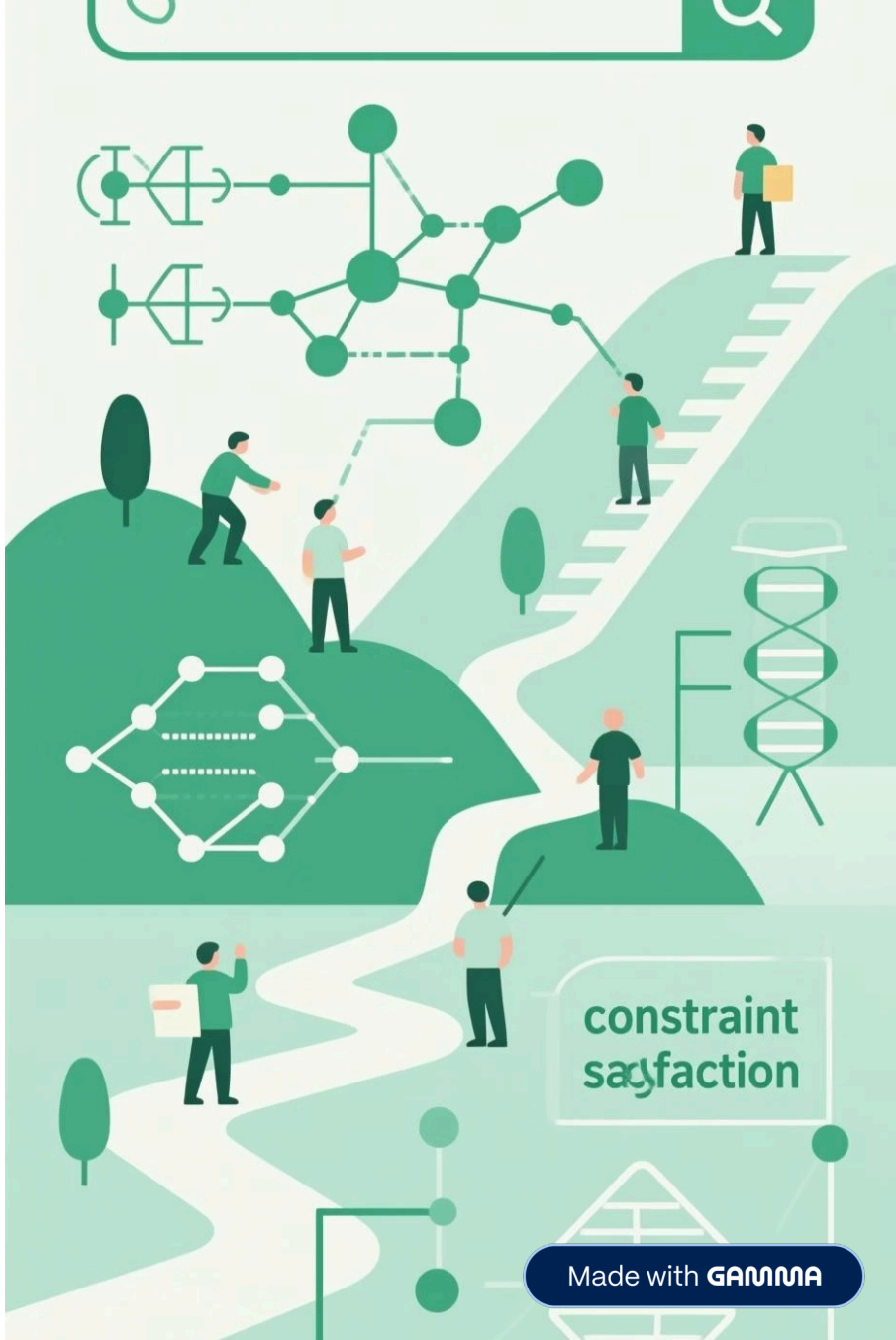
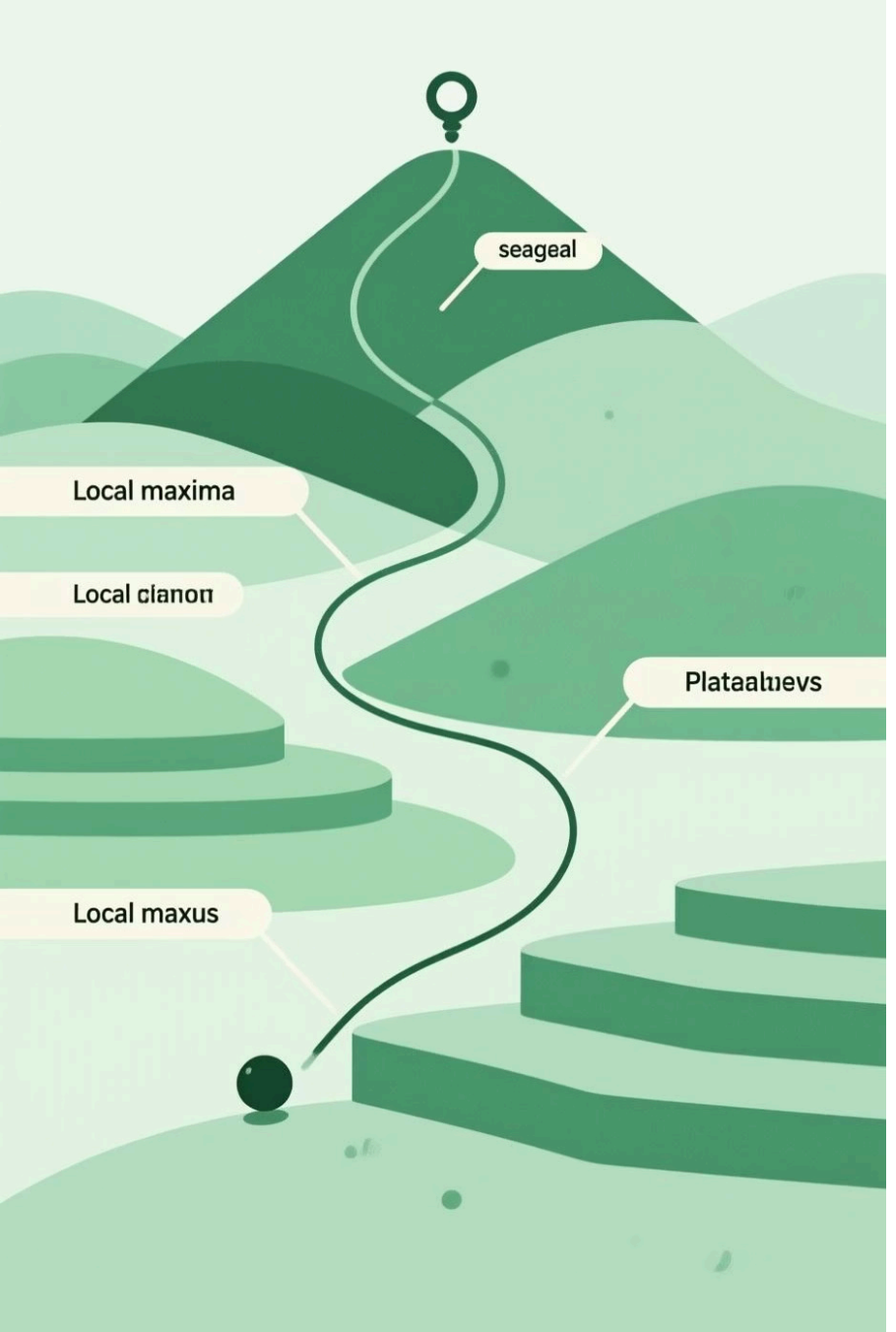
Estrategias de Búsqueda Avanzadas

Este trabajo práctico aborda tres estrategias fundamentales de búsqueda en inteligencia artificial: búsqueda local, algoritmos evolutivos y problemas de satisfacción de restricciones. Estas técnicas son esenciales para resolver problemas complejos donde los métodos tradicionales no son eficientes.





Algoritmo de Ascensión de Colinas

Mecanismo de Detención

Se detiene cuando no hay mejora en los vecinos, lo que significa que ha alcanzado un punto donde todos los movimientos posibles empeoran la solución.

Problemas

Puede quedar atrapado en máximos locales (soluciones que son mejores que sus vecinos pero no son óptimas globalmente) o en mesetas/terrazas (zonas donde los vecinos tienen el mismo valor).

Heurísticas en Problemas de Satisfacción de Restricciones

Ordenación de Variables

- MRV: Elige la variable con menos valores legales en su dominio
- Degree heuristic: Selecciona la variable involucrada en más restricciones

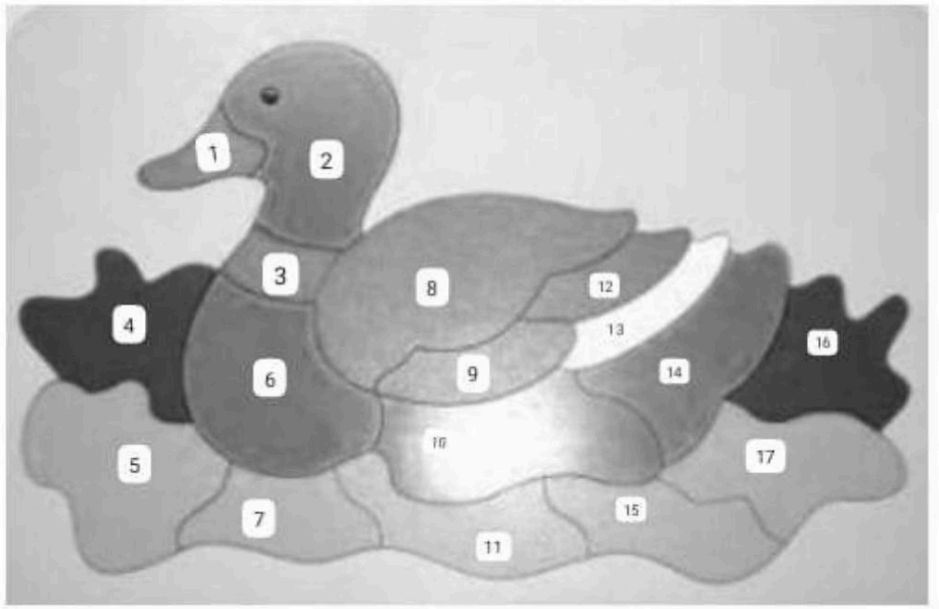
Ordenación de Valores

- LCV: Prueba primero los valores que dejan más opciones a otras variables

Inferencia durante Búsqueda

- Forward Checking: Elimina valores incompatibles de dominios vecinos
- Arc-consistency: Mantiene consistencia entre pares de variables
- MAC: Mantiene consistencia de arco después de cada asignación

Coloreo de Rompecabezas con Forward Checking



División del pato en 17 regiones numeradas para facilitar la identificación de las partes y sus vecinos.

Proceso de Coloreo

Utilizando la heurística del Valor más Restringido (MRV), se seleccionan las regiones en orden de restricción y se asignan colores verificando que no coincidan con los vecinos.

El algoritmo comienza con la región 6 (la más restringida) y progresivamente asigna colores a todas las regiones, manteniendo la consistencia mediante forward checking.

Paso	Nodo elegido (MRV)	Dominio antes	Color asignado	Vecinos afectados	Cambios en dominios
1	6	{Amarillo, Azul, Marrón, Naranja, Rojo, Verde, Violeta}	Amarillo	3, 4, 5, 7, 8, 9, 10, 11	3→Amarillo; 4→Amarillo; 5→Amarillo; 7→Amarillo; 8→Amarillo; 9→Amarillo; 10→Amarillo; 11→Amarillo
2	10	{Azul, Marrón, Naranja, Rojo, Verde, Violeta}	Azul	9, 11, 13, 14, 15	9→Azul; 11→Azul; 13→Azul; 14→Azul; 15→Azul
3	9	{Marrón, Naranja, Rojo, Verde, Violeta}	Marrón	8, 12, 13	8→Marrón; 12→Marrón; 13→Marrón
4	8	{Azul, Naranja, Rojo, Verde, Violeta}	Azul	2, 3, 12	2→Azul; 3→Azul; 12→Azul
5	11	{Marrón, Naranja, Rojo, Verde, Violeta}	Marrón	7, 15	7→Marrón; 15→Marrón
6	5	{Azul, Marrón, Naranja, Rojo, Verde, Violeta}	Azul	4, 7	4→Azul; 7→Azul
7	3	{Marrón, Naranja, Rojo, Verde, Violeta}	Marron	2	2→Azul
8	7	{Naranja, Rojo, Verde, Violeta}	Naranja		
9	4	{Marrón, Naranja, Rojo, Verde, Violeta}	Marron		
10	14	{Amarillo, Marrón, Naranja, Rojo, Verde, Violeta}	Amarillo	13, 16, 17	13→Amarillo; 16→Amarillo; 17→Amarillo
11	13	{Naranja, Rojo, Verde, Violeta}	Naranja	12	12→Naranja
12	15	{Amarillo, Naranja, Rojo, Verde, Violeta}	Amarillo	17	17→Amarillo
13	16	{Azul, Marrón, Naranja, Rojo, Verde, Violeta}	Azul	17	17→Azul
14	17	{Marrón, Naranja, Rojo, Verde, Violeta}	Marron		
15	2	{Amarillo, Marrón, Rojo, Verde, Violeta}	Amarillo	1	1→Amarillo
16	1	{Azul, Marrón, Naranja, Rojo, Verde, Violeta}	Azul		
17	12	{Amarillo, Rojo, Verde, Violeta}	Amarillo		

Búsqueda del Máximo con Hill Climbing

Función Objetivo

$f(x) = \sin(x) / (x + 0.1)$ en el intervalo [-10; -6]

Esta función tiene varios máximos y mínimos locales, ideal para demostrar las características del algoritmo.

Estrategia

1. Evaluar la función en la posición actual
2. Explorar vecinos a izquierda y derecha
3. Moverse al vecino que mejora el valor (enfoque voraz)
4. Si no hay mejora, reducir el tamaño de paso

Resultado

Mejor valor encontrado: $x = -7.723$, $f(x) = 0.1301$

El algoritmo ajusta dinámicamente el paso hasta alcanzar la precisión requerida.

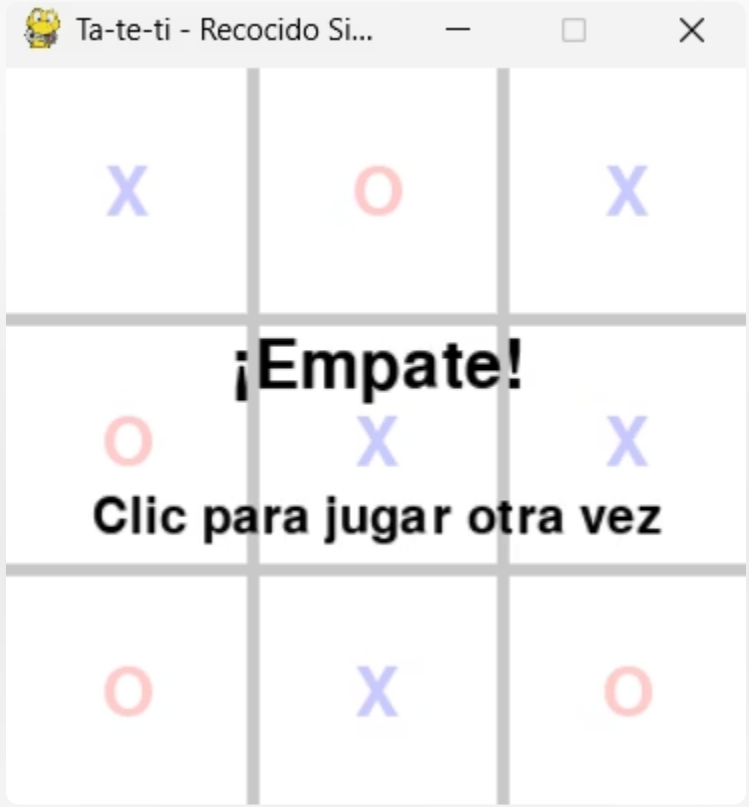
Recocido Simulado para Ta-te-ti

Características

- Implementa el algoritmo de recocido simulado para jugar al Ta-te-ti contra un humano
- Permite variar la temperatura inicial para observar diferentes comportamientos
- Incluye una función de evaluación mejorada que considera posiciones estratégicas

Efecto de la Temperatura

- Temperatura alta: Mayor exploración, decisiones más arriesgadas y variadas
- Temperatura baja: Comportamiento más conservador y predecible, pero puede quedarse en óptimos locales
- Temperatura media: Balance entre exploración y explotación



Algoritmo Genético: Problema de la Grúa

Objetivo: Maximizar el precio sin superar el límite de carga de 1000 kg

Elemento (j)	1	2	3	4	5	6	7	8	9	10
Precio (pj)	100	50	115	25	200	30	40	100	100	100
Peso (wj)	300	200	450	145	664	90	150	355	401	395

Implementación del Algoritmo Genético

Representación del Individuo

Lista binaria donde 1 indica que la caja está cargada y 0 que no lo está.

Población Inicial

Conjunto de individuos aleatorios que no superan el límite de peso de 1000 kg.

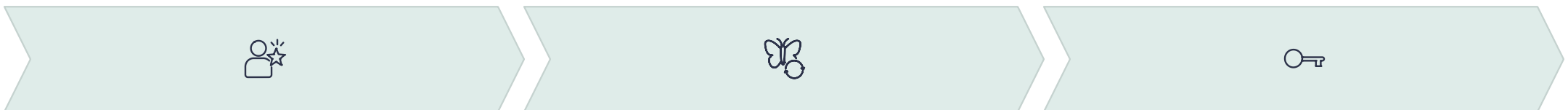
Evaluación y Selección

La función de fitness calcula el precio total. La selección se realiza mediante el método de la ruleta.

Operadores Genéticos

Cruce de un punto entre padres y mutación con probabilidad MUTATION_RATE. Si un individuo excede la capacidad, se repara eliminando cajas.

Ejecución del Algoritmo Genético



Inicialización

Crear población inicial de 20 individuos válidos

Evolución

Iterar por 100 generaciones aplicando selección, cruce y mutación

Resultado

Obtener la mejor solución encontrada con su peso y precio total

El algoritmo incluye elitismo para preservar los mejores individuos entre generaciones y muestra la evolución del fitness a lo largo del tiempo.

Conclusiones

Búsqueda Local

Los algoritmos de ascenso de colinas son eficientes pero pueden quedar atrapados en óptimos locales. El recocido simulado introduce aleatoriedad controlada para escapar de estos óptimos.

Satisfacción de Restricciones

Las heurísticas como MRV y forward checking mejoran significativamente la eficiencia al reducir el espacio de búsqueda.

Algoritmos Genéticos

Son efectivos para problemas de optimización combinatoria como el de la grúa, encontrando buenas soluciones en espacios de búsqueda grandes.