
Universidad
Tecmilenio®

Universidad Tecmilenio

Aprendizaje Automático No Supervisado

Programación dinámica y aprendizaje reforzado

Introducción

El espacio de acción es prácticamente infinito, ya que hay 32 piezas en un total de 64 casillas, en donde cada pieza tiene diferentes tipos de movimientos permitidos. Una de las estimaciones conservadoras sobre el número de movimientos posibles en el ajedrez se calcula en alrededor de 10120, valor que también se conoce como **número de Shannon**.

Una de las aplicaciones clásicas del aprendizaje reforzado es resolver el problema de ganar una partida de ajedrez. En este juego, cada movimiento que realiza cualquiera de los lados abre una nueva posición en el tablero. El objetivo final es capturar al rey del rival, pero los objetivos a corto plazo consisten en capturar también otras piezas o controlar el centro.

La supercomputadora Deep Blue, diseñada por IBM (específicamente para jugar al ajedrez), fue la primera máquina capaz de derrotar al campeón mundial de esa época: Gary Kasparov, en 1997. Esta proeza se consideró un hito en el aprendizaje automático. Desde entonces, las computadoras se han vuelto cada vez mejores en este juego, llegando a su máxima expresión con el sistema AlphaZero de Google.

En este tema aprenderás sobre el aprendizaje reforzado y parte de la teoría matemática en la que se apoya: la programación dinámica.

Explicación

Ecuación fundamental de la programación dinámica

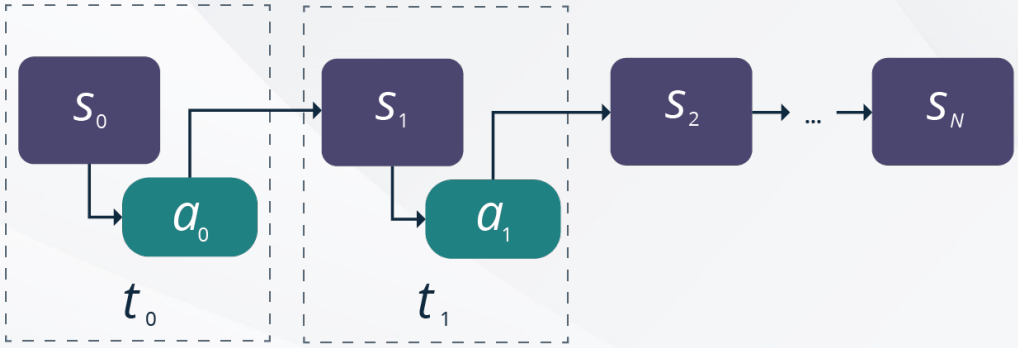
Es posible distinguir entre dos tipos principales de programación dinámica: la determinista (en la cual se utilizan datos ciertamente conocidos) y la probabilística (donde los datos se determinan a través de distribuciones de probabilidad). En ambas aproximaciones existe una noción de máquina de estados, en donde se representan las soluciones secuenciales de los subproblemas y en la que el contexto de los problemas posteriores cambia dinámicamente en función de la solución de problemas precedentes.

La parte “dinámica” de su nombre se refiere ciertamente al comportamiento variable de estas soluciones. Por ende, resolver este tipo de problemas fue uno de los motivantes de las primeras etapas de la inteligencia artificial, las cuales encontraron su forma madura con el surgimiento de los sistemas expertos (Joshi, 2019).

Explicación

Ecuación fundamental de la programación dinámica

El problema que la programación dinámica intenta resolver se puede expresar en una sola ecuación denominada **ecuación fundamental** o **ecuación de Bellman**. En la figura se puede observar gráficamente cómo se representa la configuración de la ecuación de Bellman.



Para expresarla de forma matemática se puede considerar un proceso conformado por *n* pasos, donde en cada uno existe un estado y un posible conjunto de acciones. Si se considera el estado inicial s_0 y la primera acción tomada como a_0, se puede limitar el conjunto de posibles acciones en el paso t como at $a_t \in \Gamma(s_t)$. Dependiendo de la acción realizada, se alcanza el siguiente estado.

Explicación

Ecuación fundamental de la programación dinámica

Denominando T(s, a) a la función que combina el estado actual junto con la acción y que además produce el siguiente estado s_{t+1}, entonces se puede establecer que: $s_{t+1} = T(s_t, a_t)$. Suponiendo que el problema que se está tratando de resolver implica optimizar el valor de una función $V(s_t)$ en el paso t, y que dicho proceso pasa por múltiples estados, a partir del principio de optimización iterativa, propuesto por Bellman (el cual plantea que para tener el valor óptimo en el último paso es necesario tener el valor óptimo en el paso anterior), se puede escribir lo siguiente:

$$V(s_t) = \max_{a_t \in \Gamma(s_t)} (F(s_t, a_t) + V(T(s_t, a_t)))$$

Ecuación fundamental de la programación dinámica

La expresión anterior es un caso especial de la ecuación de Bellman cuando no se considera el costo después de cada estado. Por tanto, para hacer la ecuación más genérica se puede incluir la función de costo en el paso *t* como *F* (st, at). Replanteando la ecuación de Bellman, esta quedaría de la siguiente forma:

$$V(s_t) = \max_{a_t \in \Gamma(s_t)} (F(s_t, a_t) + V(T(s_t, a_t)))$$

En algunos escenarios no se puede suponer que la optimización del valor futuro será completamente alcanzable, por lo que es necesario agregar un factor de descuento como β, donde 0 < β < 1. Entonces, la ecuación fundamental de la programación dinámica se puede escribir como:

$$V(s_t) = \max_{a_t \in \Gamma(s_t)} (F(s_t, a_t) + \beta V(T(s_t, a_t)))$$

Explicación

Aplicaciones de la programación dinámica

Una empresa que se dedica a la manufactura recibe una serie de trabajos, cuya ejecución planifica de manera mensual. Debido a una avería en uno de sus equipos principales, la junta directiva ha decidido seleccionar solamente una fracción de los trabajos pendientes para ejecutarlos, posponiendo el resto para cuando vuelvan a tener lista su máxima capacidad.

El listado de las tareas por hacer se muestra en la tabla, en donde se indica el tiempo que tarda el maquinado en cada una de las piezas y el beneficio económico que se obtiene por su terminación.

Pieza	Tiempo de maquinado (horas)	Precio (pesos)
1	4	500.00
2	3	250.00
3	10	1500.00
4	12	1200.00
5	9	1200.00
6	5	1000.00
7	6	800.00
8	7	950.00

Explicación

Aplicaciones de la programación dinámica

El problema por resolver consiste en determinar, a partir de la capacidad actual de la planta, cuál es el máximo beneficio económico que se puede obtener seleccionando adecuadamente las piezas a procesar.

La ecuación fundamental de la programación dinámica se puede representar computacionalmente para esta situación, coincidiendo con el algoritmo que se utiliza para resolver el problema de la mochila.

Pieza	Tiempo de maquinado (horas)	Precio (pesos)
1	4	500.00
2	3	250.00
3	10	1500.00
4	12	1200.00
5	9	1200.00
6	5	1000.00
7	6	800.00
8	7	950.00

Explicación

Aplicaciones de la programación dinámica

En la función principal, los parámetros v, w, y C indican los valores de los elementos (precios), los pesos (tiempo de maquinado) y la capacidad de trabajo, respectivamente.

Si consideramos que la planta tiene una capacidad de trabajo de 28 días mensuales, entonces, introduciendo los datos iniciales, se puede obtener el valor del máximo beneficio.

Pieza	Tiempo de maquinado (horas)	Precio (pesos)
1	4	500.00
2	3	250.00
3	10	1500.00
4	12	1200.00
5	9	1200.00
6	5	1000.00
7	6	800.00
8	7	950.00

Explicación

Aplicaciones de la programación dinámica

Al programar y aplicar el algoritmo se obtiene que, seleccionando de manera adecuada las piezas a maquinar, el máximo beneficio que se puede obtener es 4,250.00 pesos.

Es importante señalar que esta función únicamente permite determinar la solución óptima al problema (aunque las piezas específicas que se deberán seleccionar todavía no se conocen), por lo que será necesario desarrollar una segunda parte del algoritmo para solucionar esa situación.

Pieza	Tiempo de maquinado (horas)	Precio (pesos)
1	4	500.00
2	3	250.00
3	10	1500.00
4	12	1200.00
5	9	1200.00
6	5	1000.00
7	6	800.00
8	7	950.00

Explicación

Características del aprendizaje reforzado

El marco de aprendizaje reforzado se basa en la interacción entre dos entidades principales: el sistema y el entorno. Entre las características fundamentales que te van a permitir comprender con precisión en qué se diferencia con el resto de los métodos de aprendizaje automático podemos mencionar las siguientes:

- No hay datos de entrenamiento etiquetados preestablecidos disponibles.
- El espacio de acción está predefinido, por tanto, puede contener una gran cantidad de posibles acciones que el sistema puede realizar en cualquier instancia determinada.
- El sistema elige qué acción realizar en cada momento, por lo que el significado de instancia es diferente para cada aplicación.
- La retroalimentación se registra en todo momento (también se le suele llamar recompensa del entorno) y esta puede ser positiva, negativa o neutra.

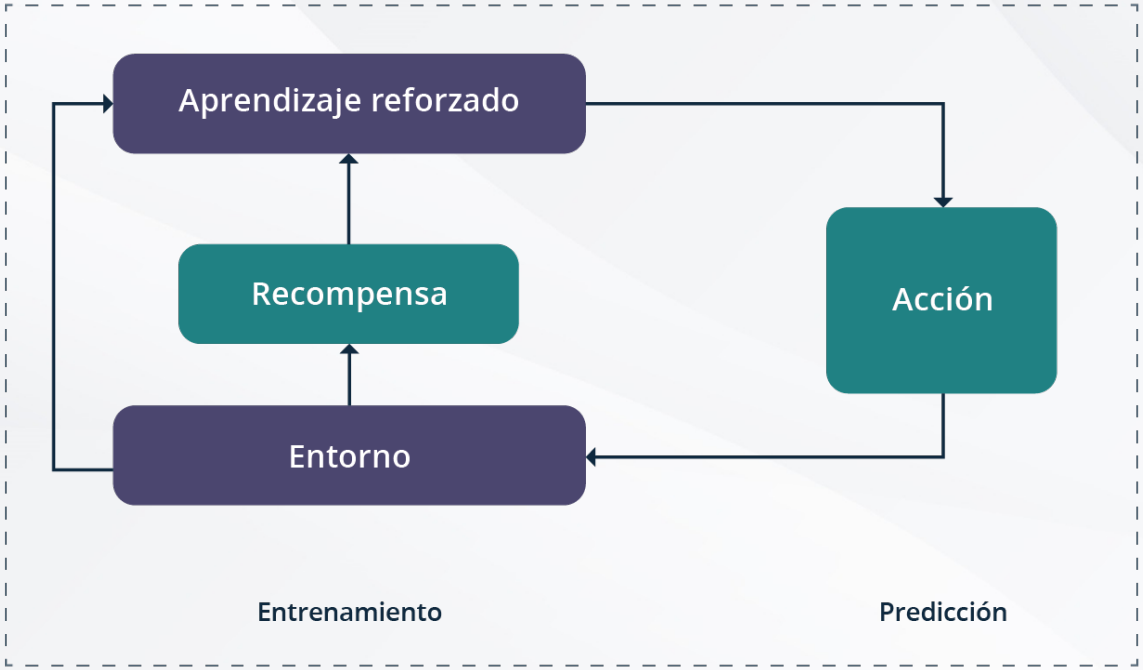
Explicación

Características del aprendizaje reforzado

- Puede haber demoras en el proceso de retroalimentación.
- El sistema aprende mientras interactúa con el entorno.
- El entorno no es estático, por lo que cada acción realizada por el sistema puede potencialmente cambiar el entorno en sí.
- Debido a la naturaleza dinámica del entorno, el espacio total de entrenamiento es prácticamente infinito.
- En este tipo de aprendizaje la fase de entrenamiento y la fase de aplicación no están separadas. El modelo está aprendiendo continuamente, por lo que también está prediciendo.

Características del aprendizaje reforzado

El aprendizaje por refuerzo es un marco de trabajo, por lo que, cualquier algoritmo que cumpla con las características antes mencionadas, se puede considerar dentro de esta familia. Para comprender de una forma más clara e intuitiva las respectivas diferencias con su homólogo de aprendizaje supervisado, en las figuras 2 y 3 se representan gráficamente sus estructuras básicas:

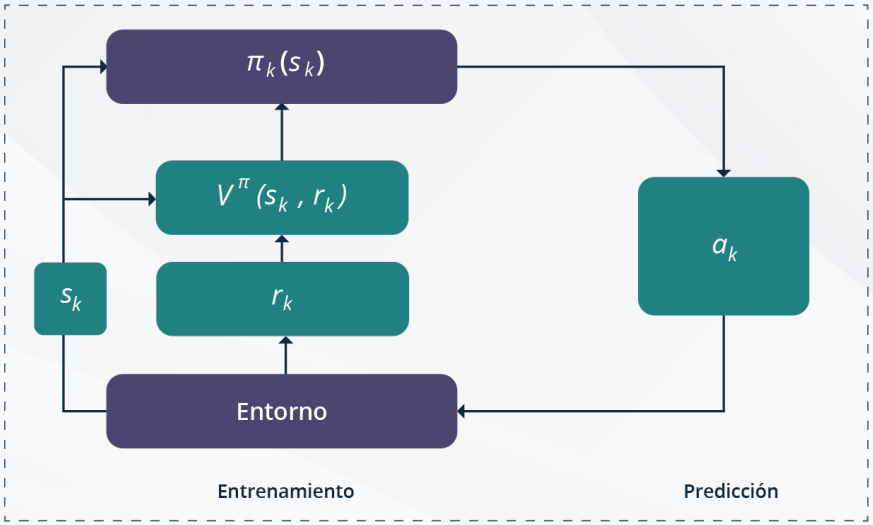


Explicación

Características del aprendizaje reforzado

El aprendizaje por refuerzo es un marco de trabajo, por lo que, cualquier algoritmo que cumpla con las características antes mencionadas, se puede considerar dentro de esta familia. Para comprender de una forma más clara e intuitiva las respectivas diferencias con su homólogo de aprendizaje supervisado, en las figuras 2 y 3 se representan gráficamente sus estructuras básicas:

El elemento a_k es la acción realizada por el sistema y r_k es la recompensa otorgada por el entorno. El término π_k se utiliza para referirse al criterio (política) que se utiliza para determinar cuál es la acción por realizar en la misma instancia de tiempo y la función del estado actual.



Por último, $V^{\pi}(s_k, r_k)$ denota la función de valor que actualiza la política utilizando el estado actual y la recompensa.

Explicación

Ejemplos de aplicaciones de aprendizaje por refuerzo

- **Robótica:** el entrenamiento de un robot para que opere en el mundo real es un problema clásico del aprendizaje por refuerzo, el cual tiene una amplia similitud con el aprendizaje biológico.
- **Videojuegos:** la resolución de videojuegos es otra aplicación muy interesante de los problemas de aprendizaje por refuerzo.
- **Personalización:** varios sitios web de comercio electrónico como Amazon o plataformas de distribución de contenidos audiovisuales como Netflix tienen la mayor parte de su contenido personalizado para cada usuario. Esto también se puede lograr con el uso de modelos de aprendizaje reforzado.

Actividad

Después de haber estudiado el tema, aborda las siguientes cuestiones:

- ¿A qué otro tipo de problema podrías aplicar el método de programación dinámica?
- Piensa en otro ejemplo de aplicación que trabaje con aprendizaje reforzado.

Cierre

En este tema se dio un recorrido por la teoría de la programación dinámica, la cual se apoya en una metodología estructurada e iterativa para descomponer un problema y resolverlo de forma secuencial. Existen dos tipos principales de programación dinámica: la determinista y la probabilística, pero ambas coinciden en la forma de representar las soluciones progresivas de los subproblemas en un contexto cambiante, utilizando la ecuación fundamental desarrollada por Bellman.

La programación dinámica está estrechamente relacionada con el aprendizaje reforzado, el cual se centra en la interacción directa con el medio ambiente, mismo que se asemeja más al aprendizaje humano que a las técnicas tradicionales de aprendizaje supervisado y no supervisado.

Finalmente, revisaste algunas de las aplicaciones más destacadas del aprendizaje reforzado, por ejemplo, la robótica, el desarrollo de videojuegos y la personalización de productos informáticos como Amazon o Netflix.

Universidad Tecmilenio

Aprendizaje Automático No Supervisado

Algoritmos evolutivos

Introducción

En la actualidad, el desarrollo de soluciones biointeligentes tiene una amplia perspectiva. Asimismo, entre sus posibles aplicaciones se encuentran la gestión de la contaminación generada por la radiación, la recolección de plásticos microscópicos, la biomedicina, entre otras.

En el desarrollo de este tema aprenderás los principios básicos que estuvieron involucrados en el desarrollo de este increíble resultado. Además, entenderás que, con el apoyo de la programación, puedes desarrollar algoritmos genéticos que te ayudarán a resolver diversas problemáticas que de otra manera serían extremadamente complejas.

Explicación

Los algoritmos evolutivos enfocan la optimización de una forma diferente, es decir, aplican la exploración masiva de posibles soluciones de una manera aleatoria, pero supervisada. Esta orientación permite el desarrollo de nuevas propuestas que, entre otras ventajas, están adecuadas de forma inherente para el procesamiento en paralelo, que es la base de la computación moderna basada en GPU.

Los algoritmos evolutivos toman como inspiración la evolución natural de los distintos sistemas biológicos, que son los fundamentos que se registraron de manera especial por la teoría de la evolución de las especies elaborada por Charles Darwin.

Explicación

La **inteligencia de enjambre** es un término general que se utiliza para referenciar a los algoritmos que están influenciados por los aspectos biológicos del comportamiento de diversos grupos de organismos primitivos, los cuales se rigen por reglas de conducta individual y grupal, siendo capaces de lograr desempeños muy complejos de manera colectiva.

El origen de las técnicas de inteligencia de enjambre se remonta a 1987, cuando Craig Reynolds publicó un artículo en donde describía este tipo de comportamiento. En su investigación, Reynolds diseñó un sistema de bandada de pájaros y asignó un conjunto de reglas para regular el comportamiento de cada una de las aves dentro del grupo.

Explicación

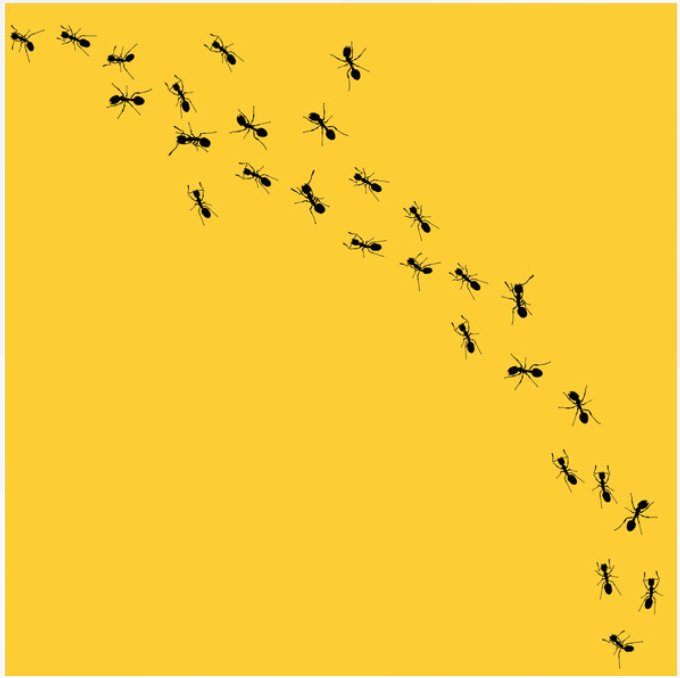
Los pasos que se necesitan para diseñar un algoritmo basado en la **inteligencia de enjambre** son los siguientes:

- Inicializar el sistema introduciendo un entorno adecuado mediante la definición de restricciones.
- Inicializar el organismo individual, definiendo las reglas de posibles acciones y formas de comunicarse con los demás.
- Establecer el número de organismos y el periodo de evolución.
- Definir las metas individuales para cada organismo y las metas grupales para todo el grupo, así como los criterios de finalización del algoritmo.
- Definir el factor de aleatoriedad que afectará las decisiones que tomen los organismos individuales al negociar entre exploración y explotación.
- Repetir el proceso hasta que se cumplan los criterios de finalización.

Explicación

La **optimización de colonias de hormigas** se puede considerar como un subconjunto de la inteligencia de enjambre, no obstante, posee algunos aspectos particulares, por lo que normalmente se estudia por separado. Los algoritmos de optimización de colonias de hormigas, como su nombre lo indica, se basan en el comportamiento de un gran grupo de hormigas en una colonia.

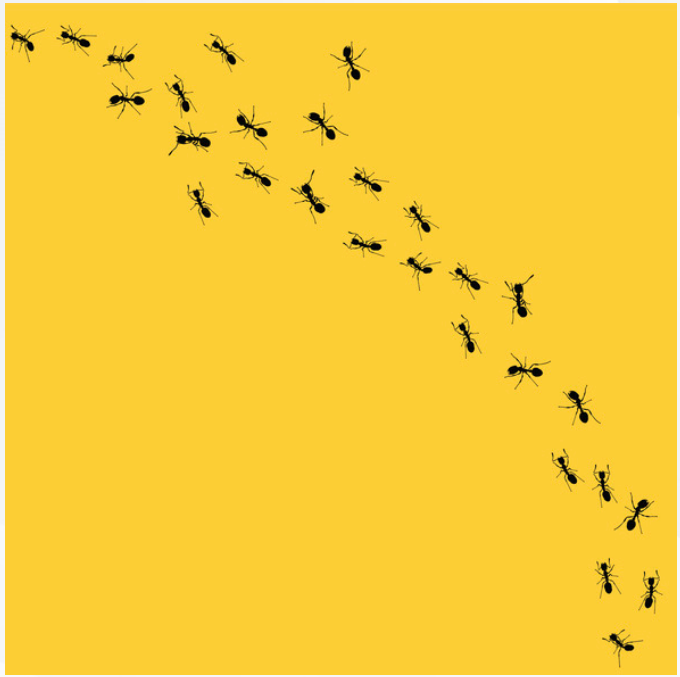
Individualmente, una hormiga posee un conjunto muy escaso de habilidades, por ejemplo, tiene una visión limitada, por lo que en casi todos los casos puede ser completamente ciega, tiene un cerebro diminuto con muy poco intelecto, y sus sentidos auditivo y olfativo tampoco son muy avanzados.



Explicación

A pesar de estas limitaciones, es bien conocido que las colonias de hormigas tienen capacidades extraordinarias, por ejemplo, la habilidad de construir nidos complejos, encontrar el camino más corto hacia las fuentes de alimento, entre otras. Cada colonia conforma un sistema completamente descentralizado, en donde no hay un tomador de decisiones central.

Por ende, todas las decisiones y acciones son decididas y ejecutadas por cada integrante en función de su propio método de funcionamiento.



Explicación

Casos de uso de los algoritmos evolutivos

- **Predecir el comportamiento de los inversores en el mercado de valores**: los consumidores que invierten en valores toman decisiones todos los días sobre si deben comprar alguna acción específica, conservar las que tienen o venderlas.
- **Selección de funciones en el aprendizaje automático:** una de las claves del aprendizaje automático consiste en generar, a partir de una serie de características sobre algún fenómeno, una conclusión que pudiera ser una clasificación o un valor numérico.
- **Descifrado de código y cifrado:** una de las aplicaciones de la informática forense que se relaciona con la seguridad de sistemas es el análisis del cifrado de mensajes codificados, los cuales se utilizan a menudo por “ciberatacantes” para ocultar información.

Explicación

Programación genética

Dentro de la programación evolutiva, los modelos de programación genética son los algoritmos más reconocidos por intentar recrear la teoría de Darwin de manera fiel. Su funcionamiento se basa en el mapeo de los conceptos de estructura genética en espacios de solución, por lo que se caracteriza por una elegante forma de implementar la selección natural y la reproducción con posibilidad de mutación de manera computacional.

Los pasos que confirman el algoritmo de programación genética se muestran en el diagrama.

Actividad

Piensa en otro problema en donde puedas aplicar la inteligencia de enjambre.

Revisa el algoritmo de programación genética e idea un plan para aplicarlo a un problema.

Cierre

En este tema conociste una nueva forma de abordar la problemática de la optimización: los **algoritmos evolutivos**. Esta propuesta toma como inspiración el desarrollo natural de los distintos sistemas biológicos. Mediante sus características de diseño en paralelo, estos algoritmos han ganado una amplia popularidad en la actualidad debido al aumento del poder computacional de los GPU modernos.

Asimismo, aprendiste sobre la inteligencia de enjambre, el cual es un tipo de algoritmo que explota muy bien el trabajo colectivo, mejorando considerablemente su desempeño en comparación con la capacidad de un elemento independiente. Del mismo modo, estudiaste los fundamentos de la optimización de colonias de hormigas que, a partir del concepto de feromonas, genera una retroalimentación positiva que actúa como una prueba de aptitud y controla la evolución en general.

Por último, conociste la esencia de los algoritmos genéticos, los cuales incluyen la selección, el cruzamiento y la mutación para construir nuevas poblaciones que sean capaces de evolucionar en cada generación.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, estas se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.
