

Descripción técnica del proyecto

Este proyecto consiste en la implementación de un sistema distribuido para la resolución de problemas de optimización utilizando algoritmos genéticos Su diseño y desarrollo integran de forma práctica los principales conceptos vistos en la asignatura de Programación Paralela y Distribuida

El sistema sigue una arquitectura de tipo **cliente-servidor** donde el servidor central actúa como coordinador y los clientes como trabajadores autónomos Se trata de una estructura **Maestro-Trabajadores** clásica en programación distribuida en la que el nodo maestro asigna tareas y recopila resultados mientras que los nodos trabajadores realizan los cálculos de forma paralela

Desde el punto de vista de la comunicación el sistema utiliza **sockets TCP** para el envío y recepción de datos entre procesos distribuidos serializados con la biblioteca pickle Esta elección permite simular un entorno distribuido incluso ejecutándose localmente y demuestra el uso práctico de técnicas de **Inter-Process Communication IPC** aplicadas a sistemas con memoria distribuida

El servidor se ha implementado de forma **asíncrona con asyncio** lo que permite gestionar múltiples conexiones de clientes concurrentemente sin necesidad de lanzar un hilo o proceso por cliente Esto reduce la sobrecarga del sistema y mejora la escalabilidad El servidor envía a cada cliente una serie de parámetros que definen el problema de optimización como el nombre de la función objetivo a minimizar el número de dimensiones el rango de búsqueda y el número de iteraciones

Por su parte los clientes están implementados como **procesos independientes** utilizando el módulo multiprocessing Cada cliente ejecuta su propio algoritmo genético con operadores de cruce mutación y selección para encontrar una solución óptima al problema planteado Cada cliente trabaja de forma autónoma y en paralelo con el resto lo que permite simular un sistema distribuido real en el que se aprovecha el paralelismo para reducir el tiempo de ejecución y explorar múltiples zonas del espacio de búsqueda

Se han definido varias funciones objetivo estándar en problemas de optimización como **Rastrigin Sphere y Rosenbrock** lo que permite evaluar el rendimiento del algoritmo genético en diferentes escenarios Estas funciones se pueden seleccionar dinámicamente desde el servidor gracias al uso de nombres y funciones como objetos lo que refuerza el diseño modular y flexible del sistema

La arquitectura del proyecto se organiza en varios módulos claramente diferenciados lo que favorece su mantenimiento y reutilización En concreto se han separado los siguientes componentes

- server contiene el servidor asíncrono y sus utilidades
- clients contiene el código del cliente y las funciones objetivo
- shared contiene los parámetros comunes de configuración
- launcher permite lanzar múltiples procesos cliente simultáneamente

Además se ha implementado un sistema automático de registro de resultados en un archivo resultados.csv donde el servidor guarda la información recibida de cada cliente

incluyendo el valor mínimo encontrado la solución correspondiente el tiempo de ejecución y la dirección del cliente Esta característica demuestra el uso de **estructuras de almacenamiento persistente** y permite analizar los resultados de forma posterior

En resumen este proyecto demuestra de forma integrada los conocimientos clave de programación paralela y distribuida incluyendo la arquitectura cliente-servidor la concurrencia con asincronía el uso de procesos paralelos la comunicación entre procesos la modularización del código y la ejecución distribuida de algoritmos de optimización Todo ello aplicado a un problema realista y con una implementación funcional completa