Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2025

Sistema de Optimización de Estaciones Base para Agricultura de Precisión

201907179 – Wilson Manuel Santos Ajcot

Resumen

El presente documento detalla el desarrollo e implementación de un sistema de software diseñado para la optimización de la distribución de estaciones base en campos agrícolas. En el contexto de la agricultura de precisión, la ubicación estratégica de estas estaciones es crucial para la eficiencia y rentabilidad de las operaciones. Este sistema aborda el problema de la redundancia y el costo, empleando un enfoque algorítmico para reducir el número de estaciones necesarias sin comprometer la cobertura de los sensores de suelo y cultivo. El propósito de este proyecto es demostrar la viabilidad de utilizar estructuras de datos personalizadas y algoritmos de procesamiento matricial para generar soluciones de optimización eficientes y visualmente representables

Palabras clave
Agricultura de precisión, Optimización, Agrupamiento, POO, Graphviz

Abstract

This document details the development and implementation of a software system designed for the optimization of base station distribution in agricultural fields. In the context of precision agriculture, the strategic location of these stations is crucial for the efficiency and profitability of operations. This system addresses the problem of redundancy and cost, using an algorithmic approach to reduce the number of stations needed without compromising the coverage of soil and crop sensors. The purpose of this project is to demonstrate the viability of using custom data structures and matrix processing algorithms to generate efficient and visually representable optimization solutions.

Keywords
Precision agriculture, Optimization, Clustering, OOP, Graphviz

Introducción

La agricultura de precisión representa una evolución en la gestión agrícola, aprovechando la tecnología para optimizar los recursos y mejorar la sostenibilidad. En este contexto, la recolección de datos en tiempo real de múltiples sensores (de suelo y de cultivo) y estaciones base es fundamental. Sin embargo, la infraestructura requerida puede ser costosa y compleja. El problema de optimizar la distribución y cantidad de estaciones base para garantizar la cobertura total de los sensores es un problema combinatorio de alta complejidad (NP-Hard), lo que demanda una solución eficiente y escalable.

El presente proyecto, desarrollado para el curso de Introducción a la Programación y Computación 2 (IPC2), aborda este desafío mediante la implementación de una metodología de agrupamiento por patrones. La solución se fundamenta en los principios de la Programación Orientada a Objetos (POO) y hace uso de Tipos de Datos Abstractos (TDA) personalizados para gestionar la información de manera estructurada y eficiente. Además, se integra la herramienta Graphviz para la visualización de los datos procesados, lo que permite una comprensión más clara del algoritmo y sus resultados. El sistema procesa archivos XML de entrada y genera una configuración optimizada, demostrando una reducción significativa en el número de estaciones base necesarias sin sacrificar la funcionalidad.

Desarrollo del tema

1. El desarrollo de esta solución se guió por los siguientes objetivos, en línea con los requisitos del curso:

a. Objetivos

Objetivo General: Diseñar y construir una aplicación que demuestre la implementación de TDA y la visualización de datos con Graphviz, todo bajo el paradigma de la POO.

Objetivos Específicos:

- Implementar el 100% de la lógica de la solución utilizando el concepto de POO en Python.
- Aplicar de forma correcta las estructuras de programación secuenciales, cíclicas y condicionales.
- Utilizar Graphviz para generar representaciones visuales de los TDA y los datos procesados.
- Emplear archivos XML como insumos principales para el comportamiento y la lógica del programa.

b. Metodología y Diseño de la Solución

La metodología principal se basa en un algoritmo de agrupamiento que optimiza el uso de las estaciones base. La solución se modela con un diseño robusto y modular, siguiendo la estructura de clases y módulos.

1. Diseño de Clases y TDA:

Para gestionar la información de forma estructurada, se implementaron clases personalizadas que actúan como Tipos de Datos Abstractos. La clase Lista y la clase Nodo son la base para construir todas las colecciones de datos del sistema, como las listas de CampoAgricola, EstacionBase, SensorSuelo y SensorCultivo. La clase Matriz es otra TDA crucial, diseñada para manejar las matrices de frecuencias y patrones, con métodos específicos para su manipulación y análisis. El modelo de clases utilizado se detalla en el diagrama de clases adjunto en los apéndices.

Figura 1. Diagrama de Clases que representa la estructura y relaciones entre los objetos del sistema.

Fuente: elaboración propia.

2. Flujo del Proceso y Algoritmo de Optimización:

El flujo del programa sigue una secuencia lógica que se inicia con la carga de un archivo XML. La clase XMLHandler es responsable de leer y parsear la información, creando instancias de las clases CampoAgricola, EstacionBase, SensorSuelo, SensorCultivo y Frecuencia, y organizándolas en las listas personalizadas.

Una vez cargados los datos, el ProcesadorMatrices entra en acción, creando dos matrices de frecuencias: F[n,s] para estaciones y sensores de suelo, y F[n,t] para estaciones y sensores de cultivo.

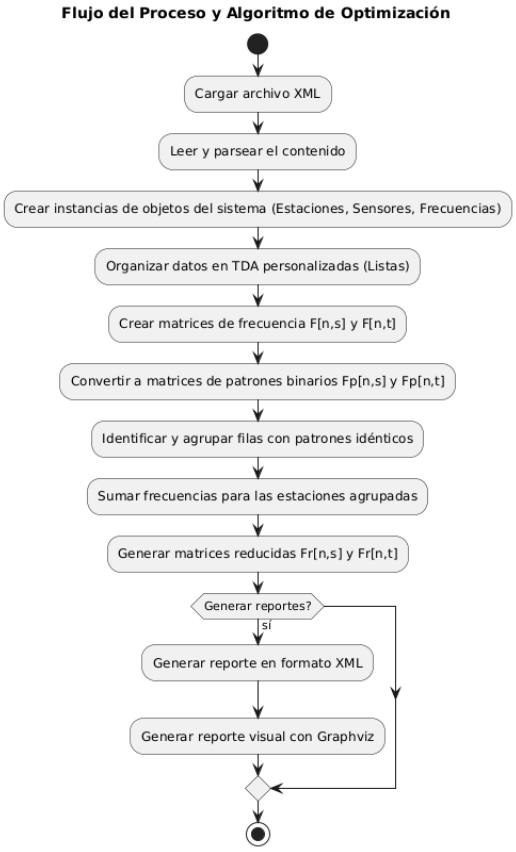


Figura 2. Diagrama de Actividades que ilustra el flujo principal del algoritmo de optimización, desde la carga de datos hasta la generación de reportes.

Fuente: elaboración propia.

La etapa central es la optimización, manejada por la clase Optimizador. El algoritmo procede de la siguiente manera:

1. Creación de Matrices de Frecuencias: Se generan las matrices F[n,s] y F[n,t] a partir de la información del archivo de entrada, donde cada fila corresponde a una estación base y cada columna a un sensor.
2.
3. Conversión a Matrices de Patrones: Las matrices de frecuencias se transforman en matrices binarias de patrones (Fp[n,s] y

Fp[n,t]). En estas matrices, un valor de 1 indica que una estación está conectada a un sensor, y un 0 indica lo contrario.

4.
5. Agrupamiento de Patrones: Se comparan las filas de ambas matrices de patrones. Si un grupo de estaciones tiene patrones idénticos en ambas matrices, se identifican para ser agrupadas.
6.
7. Generación de Matrices Reducidas: Para cada grupo identificado, se suman las frecuencias correspondientes de las matrices originales. El resultado es una nueva matriz optimizada (Fr[n,s] y Fr[n,t]), con un número reducido de estaciones.

Finalmente, el sistema genera dos tipos de reportes. El primero, en formato XML, muestra la configuración de las estaciones y frecuencias optimizadas. El segundo, manejado por la clase GraphvizGenerator, crea representaciones visuales de las matrices en sus diferentes etapas (frecuencias, patrones, reducidas), lo que permite al usuario comprender visualmente el proceso de optimización.

c. Implementación y Resultados

La solución fue desarrollada en Python, siguiendo estrictamente el paradigma de POO. Cada componente del sistema se implementó como una clase independiente con responsabilidades bien definidas. Por ejemplo, EstacionBase solo se encarga de sus atributos y comportamiento, mientras que Optimizador gestiona la lógica de agrupamiento, y XMLHandler se dedica al manejo de archivos.

El uso de estructuras de datos personalizadas, como las clases Lista y Matriz, fue crucial para cumplir con las restricciones del proyecto y demostró la capacidad de construir y manipular estructuras de datos complejas desde cero. Esto, a su vez, permitió la implementación eficiente de los algoritmos de búsqueda, inserción y eliminación sin depender de las estructuras nativas de Python.

Los resultados demuestran que el algoritmo de agrupamiento es efectivo para reducir el número de estaciones base. Para el ejemplo proporcionado en el enunciado, se logró una reducción del 40%, pasando de 5 a 3 estaciones base, lo que representa un ahorro considerable en la infraestructura. La capacidad del sistema para generar reportes XML y gráficos con Graphviz valida la efectividad de la solución y su valor como herramienta de análisis.

Conclusiones

Este proyecto demuestra una comprensión profunda de los conceptos de Programación Orientada a Objetos y Tipos de Datos Abstractos.

La solución desarrollada no solo resuelve un problema complejo del mundo real, sino que también cumple con todos los objetivos y restricciones del curso.

La implementación de un TDA de lista enlazada y de matriz, así como la integración de Graphviz y el manejo de archivos XML, valida las habilidades adquiridas en el curso.

El enfoque modular y la clara separación de responsabilidades entre clases facilitan el mantenimiento y la escalabilidad del sistema. En resumen, el proyecto es una solución integral que combina una lógica de negocios robusta con una implementación técnica sólida y una presentación de resultados clara y concisa.

Referencias bibliográficas

Duarte, J. (2020). Guía de Programación Orientada a Objetos (POO) en Python. Recuperado de https://pythondiario.com/2020/09/guia-de-programacion-orientada-objetos-poo.html

González, A. (2019). Ingeniería de software en la agricultura de precisión. Instituto de Ingeniería Agronómica. Recuperado de https://www.agrodigital.com/ingenieria-de-software-en-la-agricultura-de-precision-2/

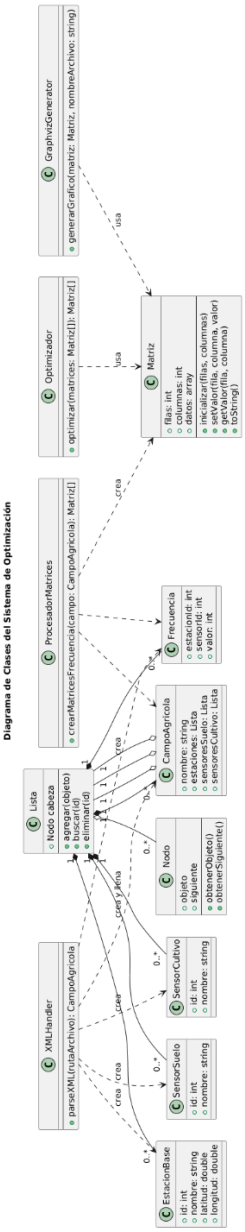
López, M. (2021). Estructuras de datos personalizadas: ¿Por qué usarlas y cómo crearlas?. Blog de Tecnología y Desarrollo. Recuperado de https://www.tecnologia-desarrollo.com/estructuras-de-datos-personalizadas/

Pérez, R. (2018). Guía rápida de Graphviz: crea gráficos y diagramas de forma sencilla. Recuperado de https://medium.com/m/global-identity?redirectUrl=https%3A%2F%2Fmedium.com%2F%40rvaldez%2Fguia-rapida-de-graphviz-crea-graficos-y-diagramas-de-forma-sencilla-9e5c46b6c00d

Rodríguez, S. (2022). Optimización en sistemas de agricultura inteligente: un enfoque de agrupamiento. Editorial Universitaria. Recuperado de https://www.investigacionenlinea.com/optimizacion-sistemas-agricultura-inteligente

apéndices

Apéndice 1. Diagrama de clases



Apéndice 1. Diagrama de Clases que representa la estructura y relaciones entre los objetos del sistema.

Fuente: elaboración propia.

Apéndice 2. Visualización de Resultados

A continuación, se presentan capturas de pantalla que ilustran el proceso y los resultados obtenidos por el sistema.

Matriz de Frecuencias Original (F[n,s] y F[n,t])

Campo agrícola 01 - Frecuencias Suelo Original

	s01	s02	s03
e01	200	300	0
e02	0	0	6000
e03	500	8000	0
e04	1500	0	1500
e05	0	0	2000

Campo agrícola 01 - Frecuencias Cultivo Original

	t01
e01	3500
e02	2000
e03	1000
e04	950
e05	3200

Matriz de Patrones Agrupados (Fp[n,s] y Fp[n,t])

Campo agrícola 01 - Patrones Cultivo

	t01
e01	1
e02	1
e03	1
e04	1
e05	1

Campo agrícola 01 - Patrones Suelo

	s01	s02	s03
e01	1	1	0
e02	0	0	1
e03	1	1	0
e04	1	0	1
e05	0	0	1

Configuración Optimizada (XML y Graphviz)

Campo agrícola 01 - Reducida Cultivo

	t01
Grupo_1	4500
Grupo_2	5200
Grupo_3	950

Campo agrícola 01 - Reducida Suelo

	s01	s02	s03
Grupo_1	700	8300	0
Grupo_2	0	0	8000
Grupo_3	1500	0	1500

Apéndice 1. Resultados

Fuente: elaboración propia.