
SISTEMA AUTOMATIZADO DE RIEGO CON DRONES

202402074 – Axel David González Molina

Resumen

Se describe el diseño e implementación de un sistema de gestión automatizada para invernaderos que utiliza drones de riego coordinados mediante estructuras de datos dinámicas propias. El sistema procesa planes de riego definidos en archivos XML y genera secuencias optimizadas de instrucciones para múltiples drones operando en paralelo. La solución implementa Tipos de Dato Abstracto (TDA) como listas simples enlazadas y colas para gestionar invernaderos, plantas, drones e instrucciones, evitando el uso de estructuras nativas de Python. El algoritmo de procesamiento paralelo reduce significativamente el tiempo total de riego al permitir que múltiples drones trabajen simultáneamente en diferentes hileras. El sistema genera reportes HTML visuales, archivos XML de salida con estadísticas y gráficos del estado de las estructuras de datos en momentos específicos utilizando Graphviz. La implementación demuestra la aplicabilidad práctica de estructuras de datos fundamentales en la resolución de problemas de optimización agrícola.

Palabras clave

Automatización agrícola
Estructuras de datos
Algoritmos de optimización
Drones de riego
Invernaderos

Abstract

This paper describes the design and implementation of an automated management system for greenhouses using coordinated irrigation drones through custom dynamic data structures. The system processes irrigation plans defined in XML files and generates optimized instruction sequences for multiple drones operating in parallel. The solution implements Abstract Data Types (ADT) such as simple linked lists and queues to manage greenhouses, plants, drones and instructions, avoiding the use of Python's native structures. The parallel processing algorithm significantly reduces total irrigation time by allowing multiple drones to work simultaneously on different rows. The system generates visual HTML reports, output XML files with statistics, and graphics of data structure states at specific moments using Graphviz. The implementation demonstrates the practical applicability of fundamental data structures in solving agricultural optimization problems.

Keywords

Agricultural automation
Data structure
Optimization algorithms
Irrigation drones
Greenhouses

Introducción

La agricultura moderna enfrenta desafíos crecientes relacionados con la eficiencia en el uso de recursos hídricos y la optimización del tiempo de trabajo. Los invernaderos agrícolas requieren sistemas de riego precisos que garanticen el suministro adecuado de agua y fertilizantes a cada planta según sus necesidades específicas.

Este proyecto presenta una solución computacional que simula y optimiza el proceso de riego automatizado mediante drones programables. El sistema permite definir planes de riego personalizados, calcular trayectorias óptimas para los drones y generar reportes detallados del proceso.

El objetivo principal es demostrar cómo las estructuras de datos fundamentales, específicamente listas enlazadas y colas implementadas desde cero, pueden aplicarse efectivamente en la resolución de problemas de optimización logística. El sistema desarrollado procesa configuraciones de invernaderos con múltiples hileras de plantas, asigna drones a cada hilera y coordina sus operaciones para minimizar el tiempo total de riego mientras mantiene el consumo eficiente de recursos.

Desarrollo del tema

Arquitectura del Sistema: El sistema se estructura en cuatro componentes principales: lectura de configuración, generación de instrucciones, cálculo de resultados y generación de reportes. La arquitectura basada en Programación Orientada a Objetos (POO) permite modelar entidades del dominio como clases independientes con responsabilidades específicas.

Las clases principales incluyen: Invernadero (contiene hileras, drones y planes), Hilera (almacena plantas en secuencia), Planta (con requerimientos de agua y fertilizante), Dron (representa un robot regador asignado a una hilera), Plan_Riego

(secuencia de instrucciones ordenadas) e Instrucción (acción específica en un tiempo determinado)

Estructuras de Datos Implementadas

El proyecto utiliza exclusivamente TDAs propios sin recurrir a estructuras nativas de Python. La `ListaSimpleEnlazada` implementa una estructura basada en nodos con punteros, permitiendo inserciones, eliminaciones y recorridos con complejidad $O(n)$. Esta estructura se utiliza para almacenar invernaderos, hileras, plantas, drones, planes e instrucciones.

La implementación incluye métodos como `agregar()`, `insertar()`, `obtener()`, `buscar_indice()`, `eliminar()` y `recorrer()`. Adicionalmente, se implementa el método `__iter__()` para permitir iteración directa compatible con Jinja2 en los templates HTML. Puede ser dividido en secciones estructurales que doten de coherencia al discurso.

Algoritmo de Optimización Paralela

El algoritmo central del sistema coordina múltiples drones operando simultáneamente. El `GeneradorInstrucciones` implementa un procesamiento paralelo que asigna tareas a cada dron según su hilera y permite que todos avancen hacia sus objetivos de forma concurrente.

El algoritmo funciona en tres fases:

1. Asignación de tareas: Cada dron recibe una cola de plantas que debe regar según el plan definido
2. Procesamiento: En cada unidad de tiempo, todos los drones ejecutan una acción (moverse, regar o esperar)

3. Finalización: El proceso termina cuando todos los drones completan sus tareas

Generación de Reportes

El sistema produce tres tipos de salidas principales. Los reportes HTML presentan tablas interactivas con instrucciones organizadas por tiempo y dron, estadísticas de consumo de recursos y visualización del proceso completo. Los archivos XML estructurados contienen tiempo óptimo, consumo total de agua y fertilizante, eficiencia por dron e instrucciones detalladas. Los gráficos Graphviz visualizan el estado de las estructuras de datos en momentos específicos del proceso.

La implementación de la interfaz utiliza Flask como framework web, permitiendo operaciones de carga de archivos XML, selección de invernadero y plan, simulación del proceso de riego, visualización de resultados en tablas dinámicas y generación de reportes descargables.

El flujo de trabajo inicia con la carga del XML que crea objetos en memoria utilizando TDAs propios. El usuario selecciona un plan específico, el sistema genera las instrucciones mediante el algoritmo paralelo, calcula estadísticas recorriendo la secuencia de instrucciones y presenta los resultados en formato HTML tabular

Conclusiones

La implementación exitosa del sistema demuestra que las estructuras de datos fundamentales, específicamente listas enlazadas, son suficientes y eficientes para resolver problemas complejos de optimización logística. El algoritmo de procesamiento paralelo reduce el tiempo de riego a aproximadamente la mitad comparado con un enfoque secuencial.

El uso de POO facilita la modularidad y mantenibilidad del código, permitiendo extensiones futuras como soporte para más tipos de acciones de drones o algoritmos de optimización alternativos. La abstracción mediante TDAs propios, aunque requiere más código inicial, proporciona control total sobre el comportamiento de las estructuras y refuerza la comprensión profunda de su funcionamiento interno.

El sistema desarrollado tiene potencial de aplicación real en agricultura de precisión, donde la optimización de recursos es crítica. Trabajos futuros podrían incluir optimización de rutas considerando obstáculos, integración con sensores IoT para monitoreo en tiempo real y algoritmos de aprendizaje automático para predecir necesidades de riego basadas en patrones históricos.

Referencias bibliográficas

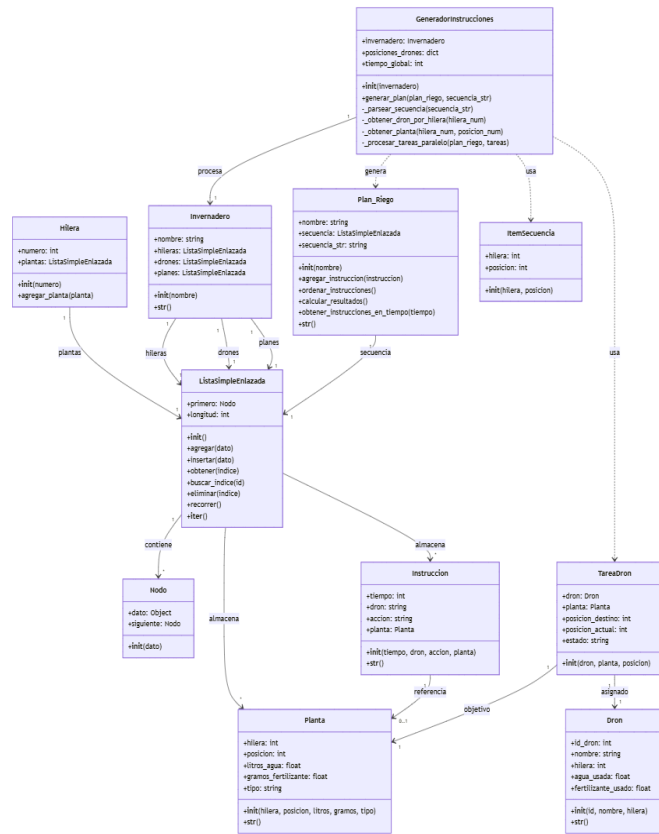
Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). **Introduction to Algorithms** (3rd ed.). MIT Press.

Sommerville, I. (2016). **Software Engineering** (10th ed.). Pearson Education.

Tanenbaum, A. S., & Wetherall, D. J. (2011). **Computer Networks** (5th ed.). Prentice Hall.

Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2013). **Data Structures and Algorithms in Python**. Wiley.

Apéndice A: Diagrama de Clases



Apéndice B: Diagrama de Actividades del Algoritmo Principal

