

Informe Proyecto II – Análisis y Diseño de Algoritmos

Laura Tatiana Coicue Poquiguet – 2276652,
 Laura Sofia Peñaloza Lopez – 2259485,
 Santiago Reyes Rodriguez - 2259738

coicue.laura, penaloza.laura, santiago.reyes.rodriguez @correounivalle.edu.co

Resumen—Este trabajo presenta un modelo matemático implementado en MiniZinc para optimizar la ubicación de nuevos programas académicos en un plano bidimensional, considerando restricciones de población, entorno empresarial y ubicaciones predefinidas.

La función objetivo maximiza la ganancia total, definida como la suma ponderada de los valores de población y entorno empresarial en las ubicaciones seleccionadas. El modelo fue evaluado utilizando los solvers Gecode, Chuffed y CBC. Los resultados muestran que Chuffed tuvo el mejor rendimiento en problemas con restricciones complejas, mientras que CBC fue más eficiente en modelos lineales. Este enfoque garantiza una distribución estratégica de los programas en función de los parámetros clave.

Abstract-- This work presents a mathematical model implemented in MiniZinc to optimize the location of new academic programs in a two-dimensional plane, considering population restrictions, business environment and predefined locations.

The objective function maximizes the total profit, defined as the weighted sum of the population and business environment values at the selected locations. The model was evaluated using the Gecode, Chuffed and CBC solvers. The results show that Chuffed had the best performance on problems with complex constraints, while CBC was more efficient on linear models. This approach ensures a strategic distribution of programs based on key parameters.

I. INTRODUCCIÓN

Este informe describe el modelo desarrollado para determinar las ubicaciones óptimas de nuevos programas académicos en un plano, respetando restricciones de población, entorno empresarial y ubicaciones predefinidas. El modelo fue implementado en MiniZinc, y se optimizó utilizando tres solvers: Gecode, Chuffed y CBC.

II. DESARROLLO DE LA SOLUCIÓN

A. Descripción

El modelo desarrollado es una representación matemática diseñada para optimizar la ubicación de nuevos programas académicos en un plano bidimensional. Utiliza una matriz $((n \times n))$ donde cada celda representa un área potencial. La

función objetivo maximiza la ganancia total, calculada como la suma ponderada de los valores de población y entorno empresarial en las celdas seleccionadas. Incluye restricciones como el mantenimiento de ubicaciones predefinidas, la prohibición de ubicaciones adyacentes y límites mínimos de población y entorno empresarial en áreas contiguas. Las soluciones se obtienen utilizando MiniZinc y los solvers Gecode, Chuffed y CBC, adaptados según las características del problema.

B. Parámetros

Dimensiones del plano:

- Número de filas y columnas de la matriz.

Ubicaciones existentes:

- Coordenadas de las ubicaciones predefinidas.
- Coordenadas de las ubicaciones predefinidas.

Matrices de entrada:

- Valor del segmento de población en la celda.
- Valor del entorno empresarial en la celda.

Restricciones:

- Valor mínimo permitido para la suma de población en las celdas contiguas.
- Valor mínimo permitido para la suma del entorno empresarial en las celdas contiguas.
- Número total de nuevos programas a ubicar.

C. Variables

- Variable binaria que indica si la celda fue seleccionada como ubicación para un nuevo programa. sí es seleccionada, en caso contrario.
- Ganancia total asociada con las ubicaciones predefinidas.
- Ganancia total considerando las nuevas ubicaciones seleccionadas.

D. Función objetivo

Esta expresión maximiza la ganancia total considerando el segmento de población y el entorno empresarial de las celdas seleccionadas.

E. Restricciones

- Solo se pueden construir n nuevas_ubicaciones
- No se puede construir en zonas adyacentes
- Segmento poblacional ≥ 25 y Entorno empresarial ≥ 20

F. Desarrollo

Se decidió desarrollar la solución con Python, debido a la facilidad de integración con minizinc. Se importaban los atributos necesarios para que el modelo corriera de forma correcta. Se debe instalar con PIP `INSTALL MINIZINC`, aparte de eso se debe tener instalado minizinc en el computador.

De igual forma, para la interfaz se utilizó la librería de ttkbootstrap que es una biblioteca que extiende tkinter para crear interfaces gráficas modernas y estilizadas inspiradas en Bootstrap y es ideal para diseñar GUIs más atractivas con un esfuerzo mínimo.

III. PRUEBAS

Para el desarrollo de las pruebas se crearon 3 inputs diferentes para mirar el comportamiento tanto del modelo como el de los solvers. Es de recalcar, que hasta que no se subiera un archivo válido, el modelo no empezaría a funcionar.

De igual forma, el usuario tendrá la opción de guardar la solución generada, esta también en un archivo TXT

Prueba 1: Matriz de 7x7, debía ubicar 2 nuevos puntos



Fig 1. Escoger el solver a utilizar



Fig 2. Resultado de la ejecución

Prueba 2: Matriz de 10x10, debía ubicar 5 nuevos puntos



Fig 3. Resultado de la ejecución con Gecode



Fig 4. Resultado de la ejecución con Chuffed



Fig 5. Resultado de la ejecución con CoinBC

En la ejecución de este problema, se pudo evidenciar que para problemas mas grandes, el sistema se demora más en procesarlo y solucionarlo. También se pudo ver, que el mejor solver, o el solver más rápido es el CoinBC, luego de este está el Gecode, y el solver mas lento o demorado es el Chuffed.

IV. CONCLUSIÓN

En conclusión, el proyecto presenta un modelo para la selección de nuevas ubicaciones considerando parámetros como la población, el entorno empresarial y restricciones de proximidad. Implementado en MiniZinc e integrado con Python, el enfoque combina técnicas de optimización matemática y programación para maximizar la utilidad de los recursos disponibles. De igual forma, se destaca que para algunos casos o problemas muy extensos se requiere integrar nuevas técnicas o soluciones que ayuden a que la solución sea mucho más rápida y efectiva.

REFERENCIAS

- [1] H. Wayne, "Exploring MiniZinc," Hillel Wayne Blog, [Online]. Available: <https://www.hillelwayne.com/post/minizinc/>. [Accessed: Dec. 15, 2024].
- [2] M. de Guía Ruiz Alamillos, "Generación de horarios óptimos en sistemas educativos utilizando programación por restricciones," Bachelor Thesis, Universidad Politécnica de Madrid, Madrid, 2018. [Online]. Available: https://oa.upm.es/54351/1/TFG_MARIA_DE_GUIA_RUIZ_ALAMILL_OS.pdf. [Accessed: Dec. 10, 2024].
- [3] "Accessing MiniZinc data from Python," Stack Overflow, [Online]. Available: <https://stackoverflow.com/questions/69553538/accessing-minizinc-data-from-python>. [Accessed: Dec. 5, 2024].
- [4] MiniZinc Team, "Modelamiento básico en MiniZinc," MiniZinc Documentation, [Online]. Available: <https://docs.minizinc.dev/en/stable/modelling.html>. [Accessed: Dec. 5, 2024].
- [5] MiniZinc Team, "Modelos más complejos," MiniZinc Documentation, [Online]. Available: <https://docs.minizinc.dev/en/stable/modelling2.html>. [Accessed: Dec. 20, 2024].
- [6] MiniZinc Team, "Integración con lenguajes: Python," MiniZinc Documentation, [Online]. Available: <https://docs.minizinc.dev/en/stable/python.html>. [Accessed: Dec. 7, 2024].