Proyecto 2
ADA II

CARLOS STIVEN RUIZ ROJAS, JHONY FERNANDO DUQUE VILLADA, JUAN DAVID ROJAS NARVAEZ, JUAN CAMILO DIAZ VALENCIA

RESUMEN:

En este informe se presenta el desarrollo de un modelo matemático implementado en MiniZinc para resolver el problema de ubicación de nuevos programas de ingeniería de sistemas en un país modelado como una rejilla bidimensional. El objetivo principal es maximizar la cobertura de población y el entorno empresarial, respetando restricciones geográficas y demográficas. Se incluyen ejemplos de ejecución, resultados obtenidos, y una discusión sobre la efectividad del modelo.

I. INTRODUCCION

La optimización de ubicaciones para nuevos programas académicos es una problemática clave para maximizar el impacto social y económico de la universidad del valle. Este proyecto aborda el problema mediante el uso de MiniZinc, un lenguaje para modelado y resolución de problemas de optimización. Se busca determinar las mejores posiciones en un plano cartesiano donde se puedan ubicar nuevos programas, considerando restricciones relacionadas con la distancia mínima respecto a programas existentes y niveles mínimos de población y entorno empresarial.

II. PLANTEAMIENTO DEL PROBLEMA

El problema plantea un espacio representado por una matriz de tamaño **n x n** donde cada celda contiene valores para los segmentos de población y entorno empresarial. Las ubicaciones actuales de los programas se proporcionan como coordenadas. Se deben ubicar **num_nuevas** programas adicionales cumpliendo las siguientes condiciones:

1. Los nuevos programas no deben estar adyacentes a los existentes ni entre ellos.
2. La suma del segmento de población en las celdas vecinas al programa debe ser al menos 25.
3. La suma del entorno empresarial en las celdas vecinas debe ser al menos 20.

El objetivo es maximizar la ganancia total derivada de la población y el entorno empresarial cubiertos por los programas.

III. PLANTEAMIENTO DE LA SOLUCIÓN

El modelo en MiniZinc utiliza las siguientes estructuras:

1. **Variables de decisión:**
   - nuevas: coordenadas de los nuevos programas a ubicar.
2. **Funciones auxiliares:**
   - suma_poblacion_actual y suma_entorno_actual: calculan los valores de población y entorno para los programas actuales.
   - suma_poblacion_nueva y suma_entorno_nueva: calculan los mismos valores para los nuevos programas.

3. **Restricciones:**
   - Distancia mínima entre programas.
   - Niveles mínimos de población y entorno empresarial.
4. **Función objetivo:** Maximizar la suma de población y entorno empresarial.

El modelo asegura una solución óptima utilizando un solver de MiniZinc que busca maximizar la función objetivo bajo las restricciones dadas.

IV. SOLUCIÓN IMPLEMENTADA:

- **Definición de parámetros:** Matrices de población y entorno empresarial, y coordenadas de los programas actuales.
- **Definición de variables:** Ubicaciones propuestas para los nuevos programas.
- **Restricciones:** Aseguran que se cumplan las condiciones del problema.
- **Función objetivo:** Maximiza la ganancia total derivada de las ubicaciones seleccionadas.

Restricciones

Las restricciones del modelo son las siguientes:

1. Los nuevos programas no deben estar adyacentes a los programas existentes ni entre ellos:
2. La suma del segmento de población en las celdas vecinas a cada nuevo programa debe ser al menos 25:
3. La suma del entorno empresarial en las celdas vecinas a cada nuevo programa debe ser al menos 20:

Explicación del código

La implementación del código se basa en los siguientes parámetros y lógica en el modelo:

1. **Parámetros de entrada:**
   - Estos parámetros definen las dimensiones del problema y las características iniciales:
     - **n:** Tamaño de la matriz que representa el espacio.
     - **num_actuales:** Número de programas existentes.
     - **num_nuevas**: Número de programas que se deben ubicar.
     - **actuales:** Coordenadas de los programas actuales.
     - **poblacion y entorno:** Matrices de dimensiones n x n que contienen los valores de segmento de población y entorno empresarial.
2. **Variables de decisión:**
   - **nuevas:** Coordenadas de las nuevas ubicaciones. Estas son variables de decisión que deben optimizarse para cumplir con las restricciones y maximizar la función objetivo.
3. **Funciones auxiliares:**
   - **Lógica:** Estas funciones encapsulan los cálculos necesarios para evaluar las celdas en función de las restricciones y objetivos del problema:
     - **suma_poblacion_actual(x, y) y suma_entorno_actual(x, y):** Evalúan la suma de los valores de población y

entorno empresarial para las celdas vecinas a una ubicación actual.

- **suma_poblacion_nueva(x, y) y suma_entorno_nueva(x, y):** Realizan el mismo cálculo para las nuevas ubicaciones propuestas.

- **Objetivo:** Facilitar la evaluación de restricciones y cálculos de la ganancia total.

4. **Restricciones:**
    - **Lógica:** Estas aseguran que las ubicaciones cumplan con los criterios establecidos:
        - Distancia mínima:
            - Los nuevos programas no deben estar en celdas adyacentes a las actuales ni entre ellos.
        - Niveles mínimos de población y entorno empresarial:
            - Cada nueva ubicación debe garantizar al menos 25 de población y 20 de entorno empresarial sumando las celdas vecinas.
5. **Función objetivo:**
    - **Lógica:** Maximizar la ganancia total derivada de la población y el entorno empresarial cubiertos por las nuevas ubicaciones, sumada a la ganancia de las ubicaciones actuales.
    - **Implementación:** Se calcula como la suma de suma_poblacion_nueva y suma_entorno_nueva para todas las nuevas ubicaciones, más la ganancia inicial de las ubicaciones actuales.
6. **Salida:**
    - **Lógica:** Presentar al usuario los resultados relevantes de la optimización:
        - Ganancia inicial derivada de las ubicaciones actuales.
        - Ganancia total después de incluir las nuevas ubicaciones.
        - Coordenadas de las ubicaciones actuales y las nuevas seleccionada

V. EJEMPLOS DE EJECUCIÓN:

Primer ejemplo:

```
2
2 2
5 5
7
10 10 10 10 10 10 10
10 15 15 15 15 10 10
10 15 20 20 15 10 10
10 15 20 25 20 15 10
10 15 15 20 20 15 10
10 10 10 15 15 10 10
10 10 10 10 10 10 10
5 5 5 5 5 5 5
5 10 10 10 10 5 5
5 10 15 15 10 5 5
5 10 15 20 15 10 5
5 10 10 15 15 10 5
5 5 5 10 10 5 5
5 5 5 5 5 5 5
2
```

```
450
980
2 2
5 5
3 5
4 3
```

Ingreso **Retorno**

Segundo ejemplo:

```
4
1 1
3 3
6 6
8 8
6
30 30 30 30 30 30
30 35 35 35 35 30
30 35 40 40 35 30
30 35 40 45 40 35
30 35 35 40 40 35
30 30 35 35 35 30
30 30 30 30 30 30
25 25 25 25 25 25
25 30 30 30 30 25
25 30 35 35 30 25
25 30 35 40 35 30
25 30 30 35 35 30
5
```

```
1110
3630
1 1
3 3
6 6
8 8
1 3
4 5
6 4
2 5
5 2
```

Ingreso **Retorno**

Tercer ejemplo:

```
3
3 3
4 5
6 7
7
10 10 10 10 10 10 10
10 15 15 15 15 10 10
10 15 20 20 15 10 10
10 15 20 25 20 15 10
10 15 15 20 20 15 10
10 10 10 15 15 10 10
10 10 10 10 10 10 10
10 10 10 10 10 10 10
15 15 15 15 15 15 15
15 20 20 20 20 15 15
15 20 25 25 20 20 15
15 20 25 30 25 20 15
15 20 20 25 25 20 15
15 15 20 20 20 15 15
4

```

```
855
1965
3 3
4 5
6 7
6 5
5 3
2 5
7 3
```

Ingreso **Retorno**

VI. DISCUSIÓN DE LOS RESULTADOS:

El modelo permite explorar diferentes configuraciones de entrada y asegura que las ubicaciones seleccionadas cumplan todas las restricciones impuestas. Los resultados demuestran la capacidad del modelo para maximizar las ganancias derivadas de la población y el entorno empresarial.

Ventajas:

- Solución óptima garantizada.
- Adaptabilidad a diferentes tamaños de matriz y números de programas.

Limitaciones:

- El tiempo de cálculo puede aumentar considerablemente con el tamaño de la matriz y el número de programas.
- El uso de los solvers afecta directamente el tiempo de ejecución de cada ejemplo donde gecode y chuffed fueron notoriamente más rápidos a comparación de coin-bc. Esto nos da a entender que no solo el planteamiento del problema afecta en el tiempo de solución sino el solver usado.

VII. CONCLUSION:

El modelo desarrollado en MiniZinc proporciona una solución robusta, eficiente y óptima para la ubicación de nuevos programas de ingeniería de sistemas. Al combinar restricciones específicas y una función objetivo bien definida, se logra maximizar tanto la cobertura de población como la del entorno empresarial.
La implementación demostró ser altamente adaptable, permitiendo explorar diferentes tamaños de matriz y configuraciones iniciales. Sin embargo, se observó que el tiempo de resolución depende significativamente tanto del tamaño del problema como del solver utilizado. Los solvers **gecode** y **chuffed** mostraron un rendimiento superior en comparación con **coin-bc**, lo que resalta la importancia de elegir herramientas adecuadas para problemas de optimización complejos.
A pesar de estas ventajas, una limitación clave es que el tiempo de cálculo puede aumentar significativamente en problemas de gran escala. Esto plantea la necesidad de investigar métodos de optimización adicionales o soluciones híbridas para abordar problemas aún más grandes.
En resumen, el modelo no solo cumple con los objetivos planteados, sino que también destaca por su flexibilidad y precisión. Este enfoque puede ser extendido a otros dominios, como la ubicación de servicios públicos o privados, siempre que se adapten las restricciones y los objetivos específicos al nuevo contexto.

BIBLIOGRAFÍA

[1] K. M. G. T. Peter J. Stuckey, «The MiniZinc Handbook,» [En línea]. Available: https://docs.minizinc.dev/en/stable/index.html.

[2] S. O. C. a. V. J. Varas, Optimización y modelos para la gestión., Dolmen Ediciones., 2000.

[3] P. J. S. L. D. K. a. H. S. K. Marriott, A MiniZinc Tutorial..

[4] M. T. M. R. Becket, Minizinc tute, University of Melbourne, 2007.

[5] B. V. Andrés Ramos, Modelos matemáticos de optimización, Universidad Pontificia Comillas.