PROYECTO DE ANÁLISIS Y DISEÑO DE ALGORITMOS II
Informe final proyecto: El Problema de la Planificación de Unidades de Energıa Térmica PUEnTe

Marlon Anacona, Diana Cadena, Robert Fernando Gil, Christian Villanueva.
marlon.anacona@correounivalle.edu.co, diana.marcela.cadena@correounivalle.edu.co, robert.gil@correounivalle.edu.co, christian.villanueva@correounivalle.edu.co

Universidad del Valle

Escuela de ingeniería de sistemas y computación
Universidad del valle, Cali, Colombia
21 de noviembre 2023

Modelo

El modelo propuesto, denominado PUEnTe (Planificación de Unidades de Energía Térmica) maneja un enfoque de programación lineal y mixta para abordar el problema de planificar el encendido y apagado de unidades térmicas de generación de energía eléctrica para satisfacer la demanda global de energía en un horizonte de planificación dividido en intervalos. Este se ajusta dinámicamente a las necesidades actuales y restricciones particulares que necesite la empresa, permitiendo devolver resultados coherentes con los parámetros registrados.

Detalles importantes de la implementación

- La interfaz permite un uso sencillo del programa.
- El análisis del problema para crear el modelo general con la posibilidad de encontrar la solución óptima mejora la rapidez , eficiencia y claridad a la hora de implementar las restricciones.
- El uso de las restricciones que nos da el enunciado nos ayudan a encontrar soluciones válidas y en muchos de los casos, soluciones óptimas
- El tiempo requerido para encontrar una solución óptima es proporcional al número de utpees y periodos.
- El tiempo requerido para encontrar una solución válida es proporcional al tamaño de los límites de las restricciones
- Se tienen en cuenta las restricciones sobre los rangos inferiores y superiores para mantener una solución válida u óptima al problema.

Parámetros del problema: aquellas variables que serán introducidas dentro del modelo de tal forma que esbozar la realidad a números para que el modelo pueda tener presente está simulación de realidad dentro de los cálculos. Estos parámetros son conocidos de antemano (ingresados por el operador de la planta de energía) que nos ayudarán a relacionar nuestras variables de decisión con las restricciones y la función objetivo.

J: Número de unidades de generación.
K: Número de intervalos de tiempo de planificación.
Ej: Costo de encender cada unidad.
Aj: Costo de apagar cada unidad.
Fj: Costo fijo de operación de cada unidad.
Vj: Coeficiente de costo variable para encender cada unidad.
Pj: Límite inferior de generación de energía para cada unidad si está generando.
Pj_max: Límite superior de generación de energía para cada unidad si está generando.
Supj: Límite superior de expansión de energía para cada unidad.
Infj: Límite inferior de reducción de energía para cada unidad.
P0j: Generación de energía inicial para cada unidad justo antes del horizonte de planificación.
Dk: Demanda global de energía para cada intervalo de tiempo.
Rk: Reserva de energía para cada intervalo de tiempo.
Gj: Indicador de si la unidad generó energía en el ciclo anterior.

Variable de decisión: Son los valores que determinan la solución del problema, es decir, son los valores que debemos de usar con el fin de poder cumplir el objetivo de nuestro problema.

pjk: Cantidad de energía generada por la unidad j en el intervalo de tiempo k.

Función objetivo: Es la medida que definirá la efectividad del sistema como una función matemática en nuestras variables de decisión, es decir, discrimina si una respuesta es "mala o buena". Veamos la función objetivo definida en el modelo:

Unset
```
solve minimize costo_total:
Fj[j] * UnitOperatingState[j, k] representa el costo fijo de operación de la unidad
j en el intervalo de tiempo k si está operativa.
Vj[j] * pjk[j, k] representa el costo variable asociado con la generación de
potencia por la unidad j en el intervalo de tiempo k.
Ej[j] * UnitStartup[j, k] representa el costo asociado con el inicio de la unidad j
en el intervalo de tiempo k.
Aj[j] * UnitShutdown[j, k] representa el costo asociado con apagar la unidad j en el
intervalo de tiempo k.
```

La función objetivo agrega estos costos para todos los intervalos de tiempo y todas las unidades, y el objetivo del modelo es minimizar esta suma total de costos. La palabra clave **solve minimize costo_total;** indica al solver que debe encontrar los valores de la variable de decisión (pjk) que minimizan esta función objetivo

Restricciones: Las restricciones son nuestra forma de limitar el modelo con las diferentes necesidades y características que fueron impuestas en el problema, esto con el fin de tener un resultado asociado a la realidad. Además de poder crear una región factible mucho más certera y óptima de cada instancia del problema en particular.
Para esto se implementan diferentes tipos de restricciones dentro del modelo. En la primera restricción se modela la no negatividad del generador, además de respetar el mínimo y el máximo de energía a generar se manejan variables enteras, por lo que es una restricción **entera**.

- $\forall j \in \{1, ..., J\}, \forall k \in \{1, ..., K\} : 0 \leq p_{jk} \leq P_{j_{\max}}$ y ($p_{jk} = 0$ o $p_{jk} \geq P_j$)

En la segunda restricción se maneja la demanda y la reserva del generador en cuestión, esta establece que la suma ponderada de los límites superiores de generación de todas las unidades multiplicada por su estado operativo debe ser al menos igual a la demanda actual más la reserva requerida para el período k. Como sabemos que los estados de estar operativo o no son variables binarias, esto hace que esta restricción sea **mixta**.

- $\forall k \in \{1, ..., K\} : \sum_{j=1}^{J} p_{jk} = D_k$
- $\forall k \in \{1, ..., K\} : \sum_{j=1}^{J} P_{j_{\max}} \times \text{UnitOperatingState}_{jk} \geq D_k + R_k$

Para la tercera restricción tenemos que es una restricción entera, ya que sólo está manejando el los límites de expansión y reducción de cada generador, por lo tanto es una restricción **entera**.

- $\forall j \in \{1, ..., J\}, \forall k \in \{2, ..., K\}$: Si $p_{jk} > 0$ y $p_{j,k-1} > 0$, entonces $|p_{jk} - p_{j,k-1}| \leq \text{Sup}_j$ y $|p_{j,k-1} - p_{jk}| \leq \text{Inf}_j$

Con respecto a la cuarta restricción, es nuestro identificador de estados de cada unidad (apagado, encendido, funcionando), se manejan como variables binarias junto a nuestras variables enteras, lo cual genera que nuestras restricciones sean **mixtas**

$$\begin{array}{ll} 1 & \text{si } k > 1 \text{ y } p_{j,k-1} > 0 \text{ y } p_{jk} = 0 \\ 1 & \text{si } k = 1 \text{ y } G_j = 1 \text{ y } p_{jk} = 0 \qquad (1) \\ 0 & \text{en otros casos} \end{array}$$

Por último, tenemos la restricción de la función objetivo

$$\text{costo\_total} = \sum_{k=1}^{K}\sum_{j=1}^{J}(F_j \cdot UnitOperatingState_j^k + V_j \cdot p_{jk}^k + Ej_j \cdot UnitStartup_j^k + Aj_j \cdot UnitShutdown_j^k)$$

Análisis de los árboles

El análisis de los árboles generados por su modelo para el ejemplo, y explicar sobre el cómo funcionó el mecanismo de Branch and Bound.

Para el ejemplo, se utilizaron las siguientes entradas:

J = 3;
K = 3;
Ej = [20,18,5];
Aj = [0.5,0.3,1.0];
Gj = [0,0,0];
Fj = [5,7,6];
Vj = [0.100,0.125,0.150];
Pj = [50,80,40];
Pj_max = [300,200,140];
Supj = [180,100,100];
Infj = [300,150,100];
P0j = [0,0,0];
Dk = [150,500,400];
Rk = [15,50,40];

Nodos azules: Instancias que se toman
Cuadrados rojos: Instancias incorrectas(no cumple con las restricciones)
Cuadrados verdes: Soluciones factibles
Triángulos rojos: Ramas podadas (instancias sin solución)
Rombo amarillo: Solucion optima

Los valores que se encuentran en corchetes, indica que se puede tomar un valor u otro del rango.

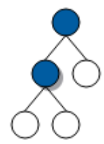
En esta primer instancia, tenemos nuestro arreglo “vacío”, ya que aún no se ha definido ninguno de sus posibles valores.

```
X_INTRODUCED_0_ = array1d(1..9, [[0..150], [160..300],
[60..300], [0..150], [80..200], [0..200], [0..140],
[40..140], [0..140]]);
costo_total = [[61..348.05]];
```

Se desglosan 2 posibles valores para nuestro vector en la última posición .

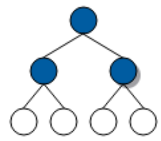
El primero, como 0.

```
X_INTRODUCED_0_ = array1d(1..9, [[0..150], [160..300],
[200..300], [0..150], [80..200], [100..200], [0..140],
[40..140], 0]);
costo_total = [[95.5..320.75]];
```

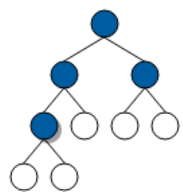


El segundo como el rango posible de valores a tomar.

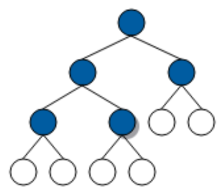
```
X_INTRODUCED_0_ = array1d(1..9, [[0..150], [160..300],
[60..300], [0..150], [80..200], [0..200], [0..140],
[40..140], [40..140]]);
costo_total = [[73..347.05]];
```



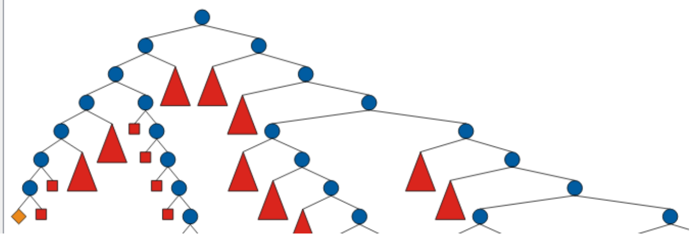
```
X_INTRODUCED_0_ = array1d(1..9, [[0..150], [160..300],
[200..300], [0..150], [80..200], [100..200], 0, [40..140],
0]);
costo_total = [[100.5..288.75]];
```



```
X_INTRODUCED_0_ = array1d(1..9, [[0..110], [160..300],
[200..300], [0..110], [80..200], [100..200], [40..140],
[40..140], 0]);
costo_total = [[112.5..306.75]];
```



```
X_INTRODUCED_0_ = array1d(1..9, [150, 300, 300, 0, 160,
100, 0, 40, 0]);
costo_total = 192.5;
```



Finalmente encuentra la solución en el rombo naranja, con los valores correspondientes a la solución óptima y su costo total.

Pruebas

Las siguientes pruebas fueron ejecutadas con el solver:
COIN-BC 2.10.10/1.17.8 en la versión de MiniZinc 2.7.6.
Adicionalmente se da la información técnica del computador que se encargó de su procesamiento:

- Procesador: Intel(R) Celeron(R) N4500 @ 1.10GHz, 1114 Mhz, 2 procesadores principales, 2 procesadores lógicos
- Memoria física instalada (RAM) 8,00 GB
- Sistema operativo: Windows 10 pro

Entrada	**Costo Obtenido**	**Solución**	**Tiempo en ejecución (Minutos)**	**Entrada J**	**Entrada K**
PuenTe	192.5	SOLUCIÓN ÓPTIMA	0.0435	3	3
PUEnTe1	608.0	SOLUCIÓN ÓPTIMA	0.05	5	5
PUEnTe2	678	SOLUCIÓN ÓPTIMA	0.02	15	10
PUEnTe3	2354.65	SOLUCIÓN ÓPTIMA	0.57	30	32
PUEnTe4	2144.65	SOLUCIÓN ÓPTIMA	2.10	50	48
PUEnTe5	3184.65	SOLUCIÓN ÓPTIMA	42.13	50	72
PUEnTe6	3430.85	SATISFECHO	50.25	80	72

PUEnTe7	3344.4	SOLUCIÓN ÓPTIMA	4.43	70	72
PUEnTe8	2144.6	SOLUCIÓN ÓPTIMA	8.42	50	48
PUEnTe9		INSATISFACIBLE	0.04	5	5
PUEnTe10		INSATISFACIBLE	0.56	50	60
PUEnTe11	616.17	SOLUCIÓN ÓPTIMA	0.04	5	5
PUEnTe12		INSATISFACIBLE	0.32	70	60
PUEnTe13	628.17	SOLUCIÓN ÓPTIMA	4.32	70	60
PUEnTe14	1059..49	SATISFECHO	7.34	70	100
PUEnTe15	2220.34	SOLUCIÓN ÓPTIMA	1.40	72	72
PUEnTe16	3287.89	SATISFECHO	25.3	80	72
PUEnTe17	2268.55	SATISFECHO	7.14	60	48
PUEnTe18	2029.65	SATISFECHO	7.30	50	48
PUEnTe19	1482.5	SOLUCIÓN ÓPTIMA	0.27	30	30
PUEnTe20		NO ARROJÓ SOLUCIÓN	12.23	100	120
PUEnTe21		NO ARROJÓ SOLUCIÓN	18.07	100	144
PUEnTe22		NO ARROJÓ SOLUCIÓN	24.06	80	72
PUEnTe23	3377.10	SATISFECHO	25.21	90	75
PUEnTe24	2645.25	SATISFECHO	17.4	75	65
PUEnTe25	1066.3	SOLUCIÓN ÓPTIMA	0.1	10	10
PUEnTe26	1626.0	SOLUCIÓN ÓPTIMA	0.04	15	15

PUEnTe27	1469.6	SOLUCIÓN ÓPTIMA	0.25	30	30
PUEnTe28	2099.9	SOLUCIÓN ÓPTIMA	2.28	50	50
PUEnTe29	2912.35	SATISFECHO	8.25	80	72
PUEnTe30		NO ARROJÓ SOLUCIÓN	32.6	100	288

Resultados con instancias creadas

Entrada	Costo Obtenido	Solución	Tiempo en ejecución (Minutos)
Instancia1	678.84	SOLUCIÓN ÓPTIMA	0.5
Instancia2	3085.25	SOLUCIÓN ÓPTIMA	0.20
Instancia3	7316.0	SOLUCIÓN ÓPTIMA	91
Instancia4	843.37	SOLUCIÓN ÓPTIMA	3
Instancia5		INSATISFACIBLE	0.75

Criterios de las instancias

La creación de instancias se llevó a cabo considerando detalladamente las características de cada uno de los parámetros que conforman el problema. Se prestó especial atención a la cantidad de plantas (J) y a la duración de los periodos (K), asumiendo una proporcionalidad relativa entre ambas variables. Esta decisión se basó en el reconocimiento de que el cumplimiento de las restricciones y la satisfacción de la demanda global de energía dependen críticamente de la relación entre estos elementos.

Aspectos clave como la cantidad de plantas, los periodos de planificación, los costos asociados con la activación y desactivación de las plantas, así como los costos fijos y variables, fueron considerados meticulosamente. Además, se evaluaron los límites de potencia, las capacidades de ampliación y reducción, y el estado inicial de cada planta. La obtención de una solución específica se basa en la consideración detallada de cada restricción del problema, y la naturaleza de la solución final está intrínsecamente vinculada a la entrada proporcionada.

Con estos principios en mente, se procedió a variar los parámetros y configurar instancias con distintas características. Algunas instancias se diseñaron con un mayor número de plantas que de períodos, mientras que otras presentaban cantidades iguales o menores. Se experimentó

con los costos de encendido y apagado, destacando instancias como la Instancia3, donde se seleccionó un escenario con menos plantas que periodos, costos de encendido superiores a los de apagado, ninguna generación de energía en periodos anteriores, y costos fijos mayores que los variables. Además, se establecieron límites proporcionales para la potencia, su ampliación y reducción. En el caso de la demanda y la reserva, se asignaron valores aleatorios que aún cumplían con las restricciones del problema, generando así varias soluciones posibles, siendo una de ellas identificada como la óptima. En otro escenario, como el representado por la Instancia5, en el cual propusimos la mayor cantidad de plantas y períodos a comparación de los otros se especificó que algunas plantas debían estar generando energía en periodos anteriores, se buscó que más de la mitad estuvieran activas en algún momento. A pesar de intentar reducir la demanda, esta instancia resultó no ser satisfacible debido a la no conformidad con la segunda restricción del problema.
Pudimos darnos cuenta que el diseño estratégico de instancias abordó la manipulación consciente de parámetros para explorar diversos escenarios y comprender cómo afectan a la solución del problema. La sensibilidad del modelo a diferentes configuraciones de entrada se destacó mediante la creación de instancias que reflejaban condiciones variadas y específicas.

Análisis de los resultados

Aprovechando los resultados obtenidos en la batería de pruebas, es crucial destacar que el código actual es capaz de alcanzar la solución óptima esperada para cada problema. No obstante, esto no implica necesariamente que el programa sea eficiente. En particular, muchas de las pruebas ejecutadas presentaron tiempos de ejecución prolongados, y algunas solo se consideraron "satisfactorias" por su tiempo de respuesta. Identificamos dos razones principales para esta situación. En primer lugar, se utilizó un computador cuyas características limitaban la velocidad del solver. En segundo lugar, se experimentaron dificultades en el modelado de las restricciones. Observamos que el problema puede ser modelado de diversas maneras, y todas estas aproximaciones pueden conducir a una solución idéntica. Esto implica la posibilidad de que nuestro modelo no sea necesariamente eficiente en cuanto a su lógica de programación de restricciones.

Adicionalmente, es fundamental enfatizar que algunos resultados obtenidos discrepan en cuanto a la cantidad de entradas de los datos en cuestión. Por ejemplo, PUEnTe6 y PUEnTe7, cuyos tiempos de ejecución son esencialmente diferentes a pesar de tener datos de entrada similares en ambos problemas. Esta variación puede atribuirse a la naturaleza intrínseca del problema, donde ciertas instancias provocan que una mayor cantidad de restricciones se activen en diferentes unidades J, lo que hace que el cálculo sea considerablemente más complejo en comparación con los resultados opuestos, donde las restricciones no se activan y se permite llegar al resultado de manera más directa.

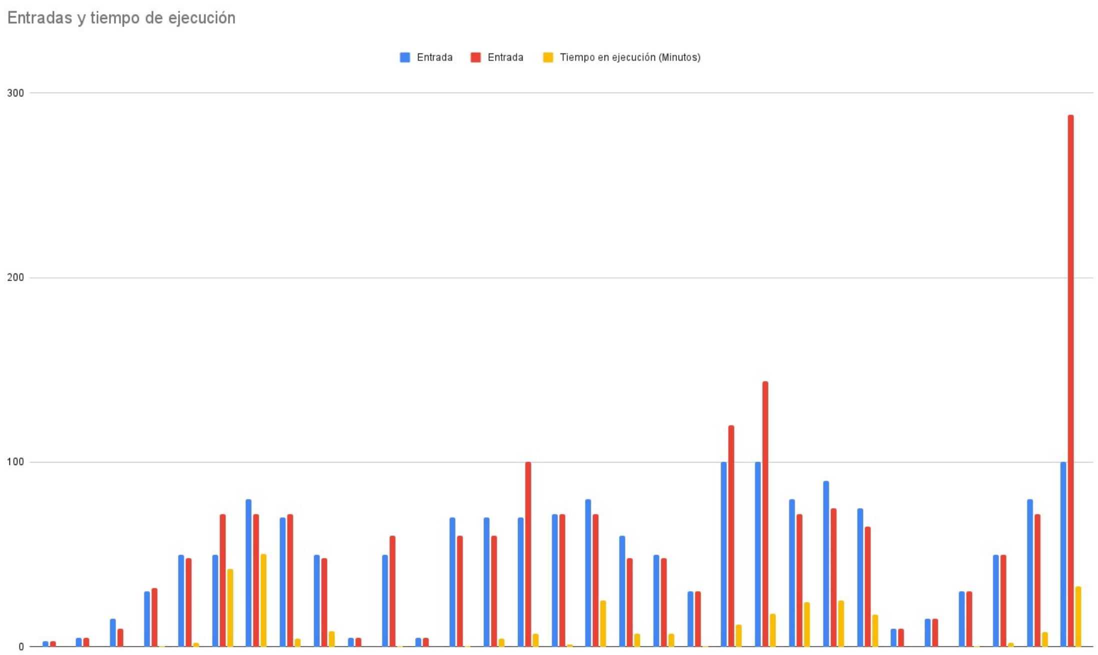


Gráfico 1. *Muestra las entradas, azul (J), rojo (k) y amarillo(tiempo de ejecución en minutos), resultados obtenidos de la batería de pruebas.*

Al elaborar el modelo, realizar las pruebas proporcionadas y crear cinco nuevas instancias para el problema, pudimos analizar un poco mejor qué parámetros tienen mayor impacto en la resolución del problema. Si bien en un principio hicimos mucho énfasis en J y K, al realizar todo lo anterior pudimos darnos cuenta de que, a la hora de realizar el cálculo del costo total y minimizarlo teniendo en cuenta las restricciones, los parámetros con mayor impacto pueden ser aquellos que están relacionados con los costos fijos y variables, ya que estos son los factores que más influyen en el costo total del problema. Otros parámetros importantes pueden ser aquellos que están relacionados con la capacidad de producción, la demanda de energía y la disponibilidad de recursos. Sin embargo, todos están relacionados, por lo que al final, todos tienen importancia; realizando alteraciones significativas a un par de ellos se obtienen resultados variables.

Vídeo

Enlace al vídeo explicativo de la interfaz
Vídeo

Conclusiones

Observando los resultados previamente presentados, podemos confirmar que nuestro algoritmo para cálculos del problema de PUEnTe es solucionado efectivamente. Sin embargo, esto no significa que sea eficiente, particularmente en situaciones más realistas donde los resultados necesitan ser de cantidades considerablemente más grandes y específicas, se debe tener aún más precisión y facilitar sus resultados en tiempos prudentes. En nuestro caso esto es una desventaja ya que el modelado que generamos no es lo suficientemente eficiente en términos de tiempo ni tampoco manejando grandes volúmenes de datos.

En conclusión, la programación lineal y mixta nos brindan herramientas extremadamente poderosas en la resolución de problemas de optimización discreta, donde premia el análisis intuitivo de las variables y parámetros de decisión