**Actividades #2 Taller-Proyecto**

Tabla de pruebas

**GRAFO 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |
| Grafo:    TPM:  [  [0, 0, 0],  [0, 0, 1],  [1, 0, 1],  [1, 0, 0],  [1, 1, 0],  [1, 1, 1],  [1, 1, 1],  [1, 1, 0]  ]  CM:  [  [0, 0, 1],  [1, 0, 1],  [1, 1, 0] | A | Cut\_one\_approximation | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0  Tiempo:  15.907953 s  Función de eficiencia:2n | MIP:  NullCut((0, 1, 2, 3))  Phi:  Φ = 0.0  Tiempo:  0.003988 s  Función de eficiencia: |  | |
| A | “no new concepts” | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0  Tiempo:  15.907953 s  Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0  Tiempo:  4.233621 s  Función de eficiencia |  |
| O | Memoization and caching | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]    Phi:  Φ = 0    Tiempo:  15.907953 s    Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0  Tiempo:  9.783897 s  Función de eficiencia: |
|  | O | Connectivity optimizations | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]    Phi:  Φ = 0    Tiempo:  15.907953 s    Función de eficiencia:2n | MIP:  NullCut((0, 1, 2, 3))  Phi:  Φ = 0.0  Tiempo:  0.001996 s  Función de eficiencia: |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]    Phi:  Φ = 0    Tiempo:  15.907953 s    Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0  Tiempo:  7.839227 s  Función de eficiencia: |  |

**GRAFO 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |
| Grafo: | A | Cut\_one\_approximation | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.612021 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.336102 s  Función de eficiencia: |  | |
| A | “no new concepts” | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.612021 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.883859 s  Función de eficiencia |  |
| O | Memoization and caching | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.612021 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.602426 s  Función de eficiencia: |
|  | O | Connectivity optimizations | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.612021 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 2.680553  Tiempo:  1.458148 s  Función de eficiencia: |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.612021 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  1.186331 s  Función de eficiencia: |  |

**GRAFO 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |
| Grafo: | A | Cut\_one\_approximation | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  1.102348 s  Función de eficiencia:2n | MIP:  Cut [B, C] ━━/ /━━➤ [A]  Phi:  Φ = 0.520834  Tiempo:  0.524636 s  Función de eficiencia: |  | |
| A | “no new concepts” | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  1.102348 s  Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C]  Phi:  Φ = 0.520834  Tiempo:  0.558397 s  Función de eficiencia |  |
| O | Memoization and caching | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  1.102348 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  0.637821 s  Función de eficiencia: |
|  | O | Connectivity optimizations | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  1.102348 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.659722  Tiempo:  0.820733 s  Función de eficiencia: |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  1.102348 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  0.565997 s  Función de eficiencia: |  |

**GRAFO 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |
| Grafo:  Diagrama  Descripción generada automáticamente | A | Cut\_one\_approximation | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  8.165864 s  Función de eficiencia:2n | MIP:  Cut [A, C, D] ━━/ /━━➤ [B]  Phi:  Φ = 0.308035  Tiempo:  1.671084 s  Función de eficiencia: |  | |
| A | “no new concepts” | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  8.165864 s  Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  1.679061 s  Función de eficiencia |  |
| O | Memoization and caching | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  8.165864 s  Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  1.679061 s  Función de eficiencia: |
|  | O | Connectivity optimizations | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  8.165864 s  Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.401785  Tiempo:  5.246937 s  Función de eficiencia: |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  8.165864 s  Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  7.011258 s  Función de eficiencia: |  |

**GRAFO 5**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  |
| Grafo:  Diagrama  Descripción generada automáticamente | A | Cut\_one\_approximation | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788  Tiempo:  61.499611 s    Función de eficiencia:2n | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788      Tiempo:  5.320051 s    Función de eficiencia: |  | |
| A | “no new concepts” | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788  Tiempo:  61.499611 s    Función de eficiencia:2n | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788    Tiempo:  9.416309 s    Función de eficiencia |  |  |
| O | Memoization and caching | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788  Tiempo:  61.499611 s    Función de eficiencia:2n | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788  Tiempo:  68.05466 s    Función de eficiencia: |  |
|  | O | Connectivity optimizations | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788  Tiempo:  61.499611 s    Función de eficiencia:2n | MIP:  Cut [A, C, D, E] ━━/ /━━➤ [B]  Phi:  Φ = 0.1875  Tiempo:  44.74539 s  Función de eficiencia: |  |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788  Tiempo:  61.499611 s    Función de eficiencia:2n | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788    Tiempo:  74.070221 s  Función de eficiencia: |  |  |

**GRAFO 6**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  | |
| Grafo:  Diagrama  Descripción generada automáticamente | A | Cut\_one\_approximation | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  1104.147865 s  Función de eficiencia:2n | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  3.890247 s    Función de eficiencia: |  |
| A | “no new concepts” | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  1104.147865 s  Función de eficiencia:2n | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  74.171171 s    Función de eficiencia |  |  |
| O | Memoization and caching | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  1104.147865 s  Función de eficiencia:2n | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  918.905389 s    Función de eficiencia: |  |
|  | O | Connectivity optimizations | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  1104.147865 s  Función de eficiencia:2n | MIP:  Cut [A, B, C, D, E] ━━/ /━━➤ [F]  Phi:  Φ = 0.401718    Tiempo:  14.453184 s    Función de eficiencia: |  |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  1104.147865 s  Función de eficiencia:2n | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  1034.164633 s  Función de eficiencia: |  |  |

**GRAFO 7**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  | |
| Grafo:  Diagrama  Descripción generada automáticamente | A | Cut\_one\_approximation | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  4181.075092s  Función de eficiencia:2n | MIP:  NullCut((0, 1, 2, 3, 4, 5))  Phi:  Φ = 0.0    Tiempo:  0.000998 s    Función de eficiencia: |  |
| A | “no new concepts” | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  4181.075092s  Función de eficiencia:2n | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  1892.358562s  Función de eficiencia |  |  |
| O | Memoization and caching | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  4181.075092s  Función de eficiencia:2n | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  2394.413088s  Función de eficiencia: |  |
|  | O | Connectivity optimizations | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  4181.075092s  Función de eficiencia:2n | MIP:  NullCut((0, 1, 2, 3, 4, 5))  Phi:  Φ = 0.0  Tiempo:  0.000998 s    Función de eficiencia: |  |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  4181.075092s  Función de eficiencia:2n | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  2376.588076s  Función de eficiencia: |  |  |

**GRAFO 9**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  | |
| Grafo:  Diagrama  Descripción generada automáticamente con confianza media | A | Cut\_one\_approximation | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  480.404822 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E, F]  Phi:  Φ = 0.859372    Tiempo:  4.416378 s    Función de eficiencia: |  |
| A | “no new concepts” | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  480.404822 s    Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  20.269228 s    Función de eficiencia |  |  |
| O | Memoization and caching | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  480.404822 s    Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  490.538942 s    Función de eficiencia: |  |
|  | O | Connectivity optimizations | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  480.404822 s    Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.749999    Tiempo:  10.593902 s    Función de eficiencia: |  |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  480.404822 s    Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  480.839672 s  Función de eficiencia: |  |  |

**GRAFO 10**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  | |
| Grafo:  Diagrama  Descripción generada automáticamente | A | Cut\_one\_approximation | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  444.673053 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  147.649329 s    Función de eficiencia: |  |
| A | “no new concepts” | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  444.673053 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  16.762605 s    Función de eficiencia |  |  |
| O | Memoization and caching | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  444.673053 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  367.494237 s    Función de eficiencia: |  |
|  | O | Connectivity optimizations | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  444.673053 s    Función de eficiencia:2n | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.026402    Tiempo:  1520.111336s  Función de eficiencia: |  |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  444.673053 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  367.031683 s  Función de eficiencia: |  |  |

**GRAFO 11**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  | |
| Grafo:  Imagen que contiene competencia de atletismo, flor  Descripción generada automáticamente | A | Cut\_one\_approximation | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  486.660469 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  135.79196 s    Función de eficiencia: |  |
| A | “no new concepts” | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  486.660469 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  19.02526 s    Función de eficiencia |  |  |
| O | Memoization and caching | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  486.660469 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  434.934237 s    Función de eficiencia: |  |
|  | O | Connectivity optimizations | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  486.660469 s    Función de eficiencia:2n | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.06666    Tiempo:  1615.613557s  Función de eficiencia: |  |  |
|  | O | Parallel\_Cut\_Evaluation | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  486.660469 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  398.610314s  Función de eficiencia: |  |  |

**Parte 2. Propuestas de mejoramiento**

En este apartado ustedes podrán elegir una o varias de las estrategias para mejorar la eficiencia del problema de las particiones (MIP) y diseñar e implementar su estrategia de mejoramiento.

Se debe presentar lo siguiente:

* Estrategia (s) de PyPhi sobre la(s) que se abordará(n)
  + Se abordará la estrategia de Parallel\_Cut\_Evaluation
* Explicación de su solución de mejoramiento
  + Se uso el método de divide and conquer en el cual dividimos todo el proceso el cual es el encargado de retornar todas las posibles particiones del mechanism and purview, usando 3 métodos adicionales para disminuir el trabajo del método principal y hacer más eficiente su funcionamiento
* Macro algoritmo-(Source code)
  + <https://github.com/lucasucaldas/GraphProject>
* Tabla de resultados
* **GRAFO 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |
| Grafo: | A | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0  Tiempo:  15.907953 s  Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0  Tiempo:  7.516224 s  Función de eficiencia: |  |
|
|

* **GRAFO 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |
| Grafo: | A | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  0.612021 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 1.916665  Tiempo:  1.066230 s  Función de eficiencia: |  |
|
|

* **GRAFO 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |
| Grafo: | A | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  1.102348 s  Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C]  Phi:  Φ = 0.520834  Tiempo:  0.432971 s  Función de eficiencia: |  | |
|
|

* **GRAFO 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |
| Grafo:  Diagrama  Descripción generada automáticamente | A | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  8.165864 s  Función de eficiencia:2n | MIP:  Cut [A, B, C] ━━/ /━━➤ [D]  Phi:  Φ = 0.308035  Tiempo:  6.315788 s  Función de eficiencia: |  |
|
|

* **GRAFO 5**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  |
| Grafo:  Diagrama  Descripción generada automáticamente | A | | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788  Tiempo:  61.499611 s    Función de eficiencia:2n | MIP:  Cut [A, B, C, D] ━━/ /━━➤ [E]  Phi:  Φ = 0.048788    Tiempo:  73.865130 s    Función de eficiencia: |  | |
|  |

* **GRAFO 6**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  |
| Grafo:  Diagrama  Descripción generada automáticamente | A | | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  1104.147865 s  Función de eficiencia:2n | MIP:  Cut [A, B, C, E, F] ━━/ /━━➤ [D]  Phi:  Φ = 0.250981    Tiempo:  1032.875204s    Función de eficiencia: |  |
|  |
|  |

* **GRAFO 7**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | Clase de mejora | | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  |
| Grafo:  Diagrama  Descripción generada automáticamente | A | | Parallel\_Cut\_Evaluation | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  4181.075092s  Función de eficiencia:2n | MIP:  Cut [D, F] ━━/ /━━➤ [A, B, C, E]  Phi:  Φ = 4.805374    Tiempo:  2375.698527s    Función de eficiencia: |  |
|  |
|  |

* **GRAFO 9**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | | Clase de mejora | | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  |
| Grafo:  Diagrama  Descripción generada automáticamente con confianza media | | A | | Parallel\_Cut\_Evaluation | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  480.404822 s    Función de eficiencia:2n | MIP:  Cut [A, B] ━━/ /━━➤ [C, D, E, F]  Phi:  Φ = 0.531249    Tiempo:  480.547253 s    Función de eficiencia: |  |
|  |
|  |
|  |
|  |

* **GRAFO 10**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | | Clase de mejora | | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  |
| Grafo:  Diagrama  Descripción generada automáticamente | | A | | Parallel\_Cut\_Evaluation | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  444.673053 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.003057    Tiempo:  366.435743 s |  |
|  |
|  |
|  |
|  |

* **GRAFO 11**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entrada de datos | | Clase de mejora | | Descripción estrategia PyPhi | Resultado Original | Resultado con  MEJORA | Observaciones |  |
| Grafo:  Imagen que contiene competencia de atletismo, flor  Descripción generada automáticamente | | A | | Parallel\_Cut\_Evaluation | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  486.660469 s    Función de eficiencia:2n | MIP:  Cut [A] ━━/ /━━➤ [B, C, D, E]  Phi:  Φ = 0.009583    Tiempo:  398.418324s    Función de eficiencia: |  |
|  |
|  |
|  |
|  |