



Presentación Final

2D Strip Packing Problem

27 de noviembre de 2023

Florencia Ramírez Sancristoful

Representaciones

01

Rectángulos

Representado con un *struct*.

02

Posiciones

Representado con un *struct*.

03

Orden

Representado con un *vector*.

04

Región

Representado con una *clase*.

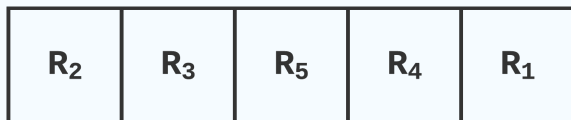


Representaciones



Rectangle 2
 $id = 2$
 $w = 4$
 $h = 6$
 $position = \{\}$

$N = 5$



$i =$

1

2

3

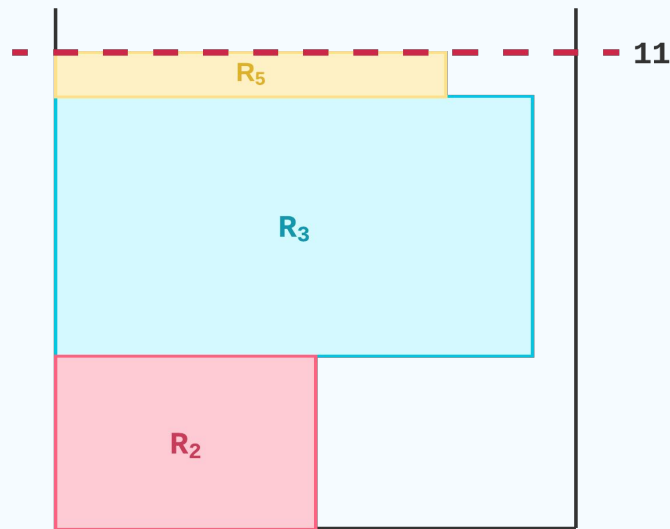
4

5

Rectangle 2
 $x = 0$
 $y = 0$
 $r = 1$

Rectangle 3
 $x = 0$
 $y = 4$
 $r = 0$

Rectangle 5
 $x = 0$
 $y = 10$
 $r = 0$



$W = 12$

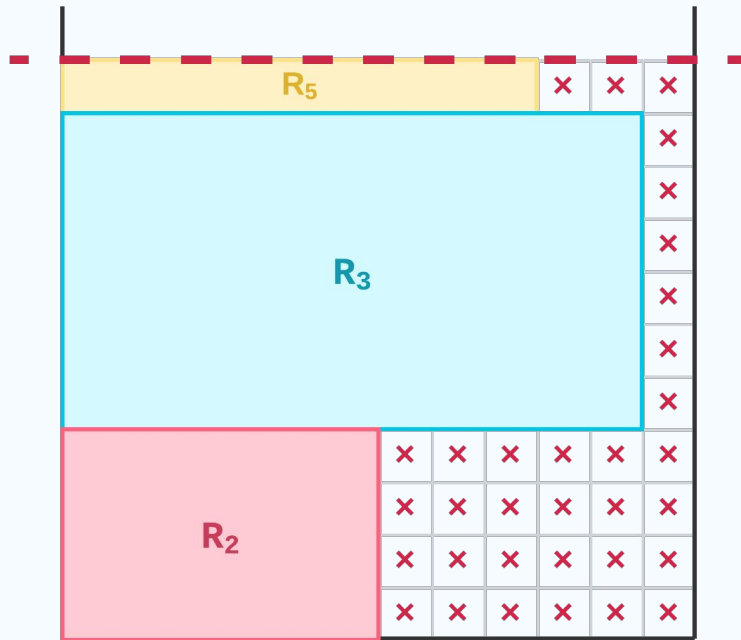
Solución inicial

```
void Strip::addRectangleGreedy(Rectangle rectangle) {
    int bestY = INT32_MAX;
    int bestX = 0;
    int orientation;
    if (rectangle.width < rectangle.height && rectangle.height <= fixedWidth) {
        orientation = 1;
        for (int x = 0; x <= fixedWidth - rectangle.height; x++) {
            int y = 0;
            for (const Rectangle &r : rectangles) {
                // Revisa si hay rectángulos posicionados en ese X
                // Obtiene el Y más alto donde se podría colocar
            }
            if (y < bestY && canBePlaced(rectangle, x, y, orientation, rectangles)) {
                // Actualiza los valores si es mejor
            }
        }
    } else {
        orientation = 0;
        // Sigue la misma lógica que el anterior
    }
    rectangle.position = {bestX, bestY, orientation};
    rectangles.push_back(rectangle);
}
```

Movimiento

Cambiar la posición de un rectángulo.

- En el peor caso, la cantidad de comparaciones depende de la **función de evaluación**.



Hill Climbing Alguna Mejora

Obtener las
posiciones
disponibles.

Vacíos



Rectángulo

Seleccionar un rectángulo para
reposicionar.



Posicionar el rectángulo en
una posición disponible.

Cambio



Revisión

Revisar si no se solapa y se
obtiene una mejor función de
evaluación.



Si se mejora y no se solapa,
volver al paso 2, sino, volver
al paso 3.

Repetición



Término

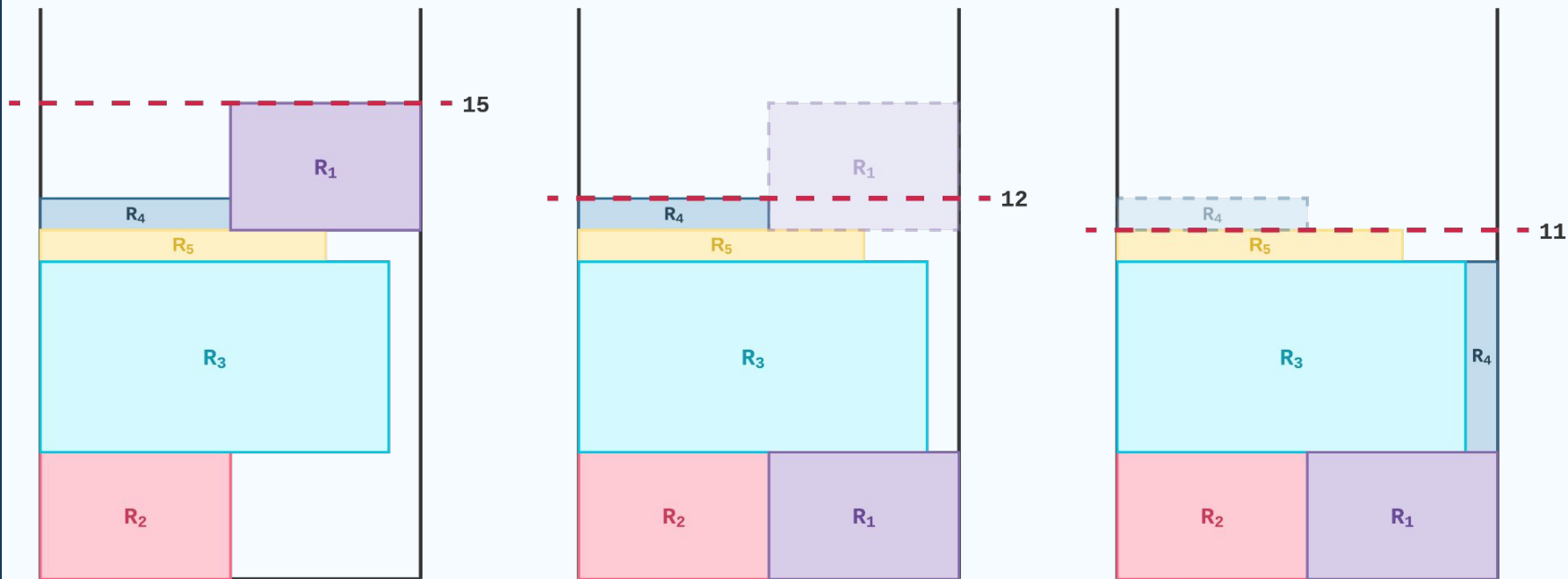
Quando revisó todos
o no se pudo
mejorar, termina.



Hill Climbing Alguna Mejora

```
void Strip::hillClimbingFirstImprovement() {
    calculateEmptySpaces();
    vector<Rectangle> improvedRectangles = rectangles;
    Rectangle currentRectangle;
    int currentSolution = evaluationFunction(rectangles);
    int improvedSolution;
    int counter = rectangles.size() - 1;
    bool local = false;
    bool improvement = false;
    while(!(local) && (counter >= 0)) {
        improvement = false;
        for (int i = 0; i < emptySpaces.size(); i++) {
            improvedRectangles.erase(improvedRectangles.begin() + counter);
            if (canBePlaced( [...] ) && [no sobrepasa el límite de la región sin rotar] ) {
                // Agrega el rectángulo en esa posición sin rotar
                // Calcula la función de evaluación
                if (improvedSolution < currentSolution) {
                    //Actualiza su posición en el vector original
                    improvement = true;
                }
            } else if (canBePlaced( [...] ) && [no sobrepasa el límite de la región rotado] ) {
                // Agrega el rectángulo en esa posición rotado
                // Sigue la misma lógica que el anterior
            }
            improvedRectangles = rectangles;
            if (improvement) {
                break;
            }
        }
        if (!(improvement)) {
            local = true;
        }
        counter--;
        emptySpaces.clear();
        calculateEmptySpaces();
    }
}
```

Hill Climbing Alguna Mejora



Experimentos



Instancias Conocidas

Utilizando los casos de prueba **BENG**, con tiempos de ejecución conocidos.

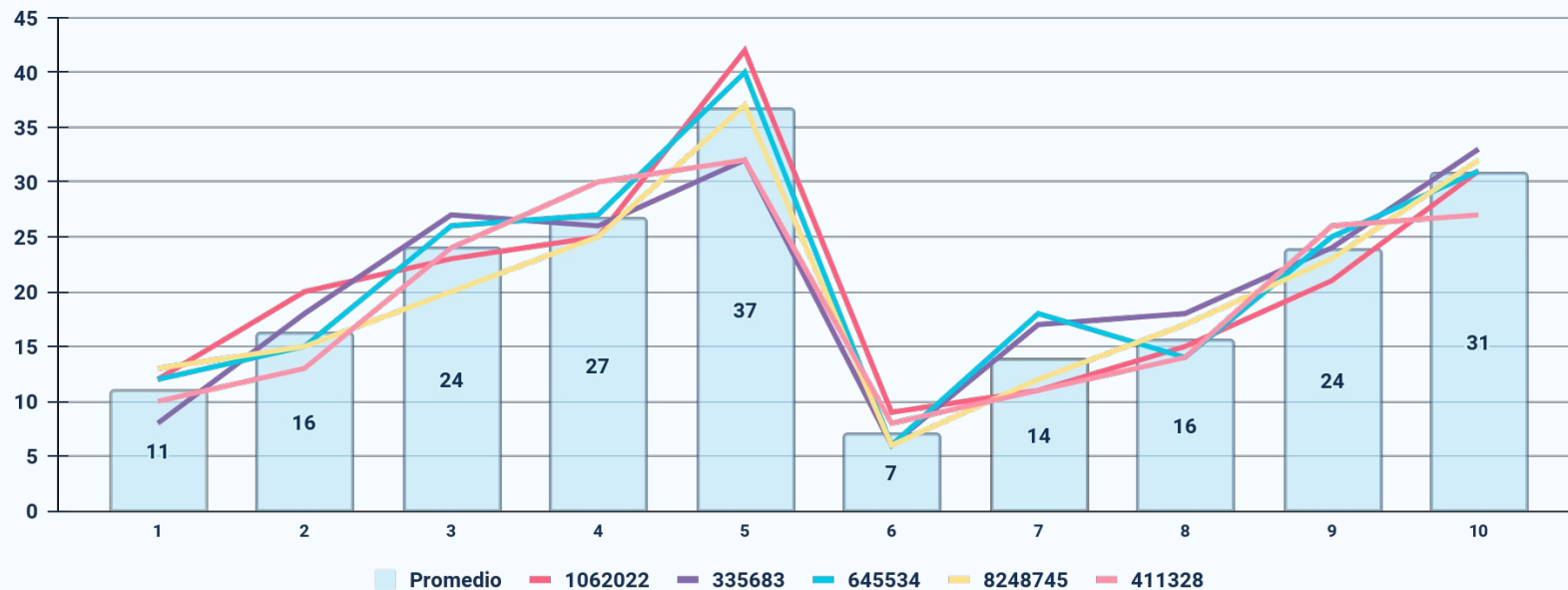


Instancias de Prueba

Utilizando los casos de prueba entregados, comparando mejoras aplicando **HC-FI**.

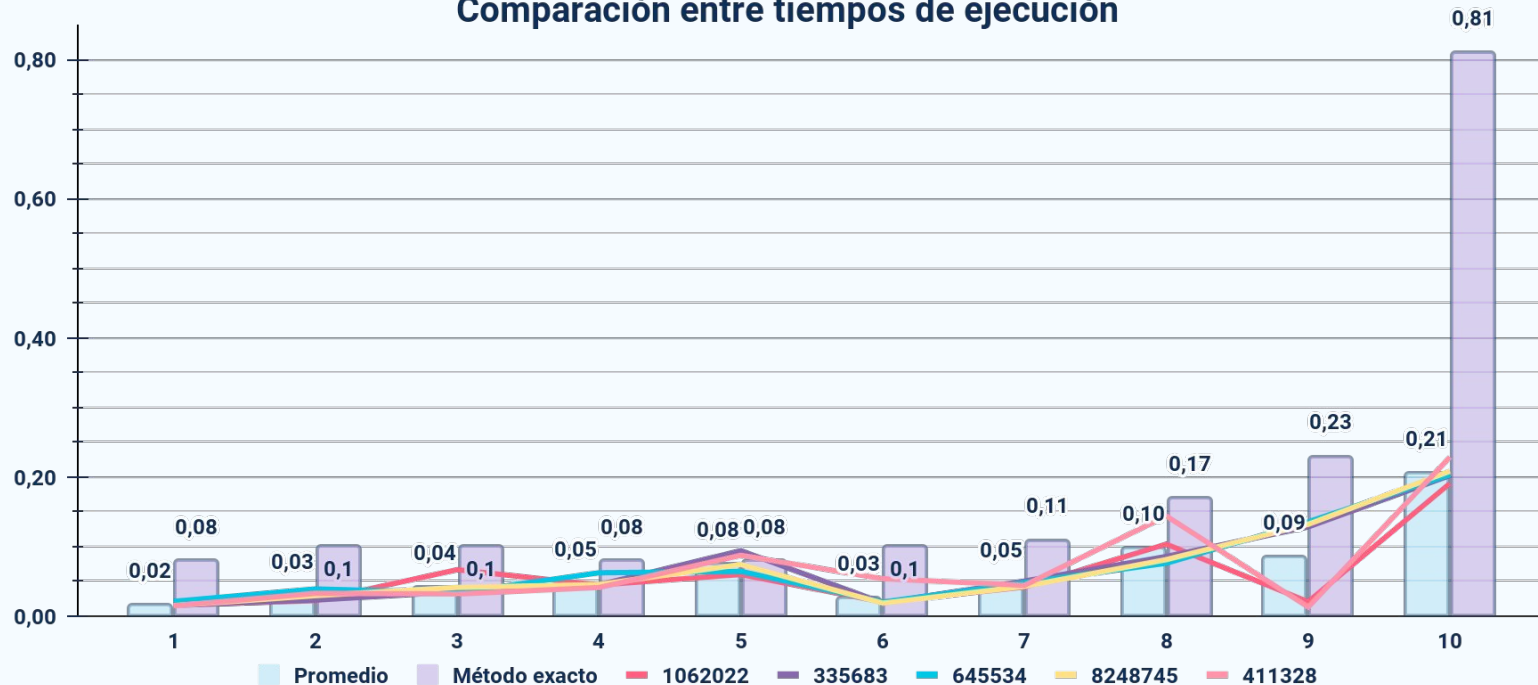
Resultados Instancias Conocidas

Diferencia entre la solución encontrada y la solución óptima



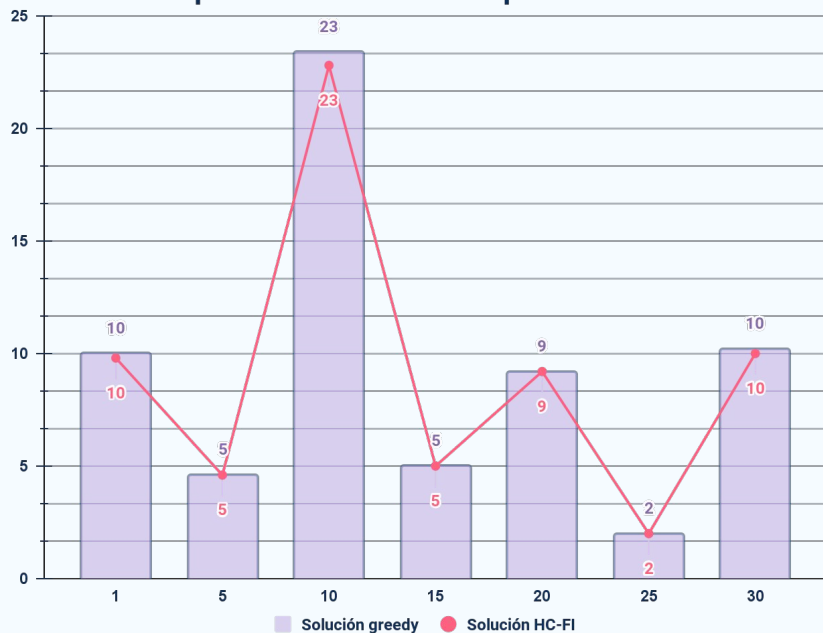
Resultados Instancias Conocidas

Comparación entre tiempos de ejecución

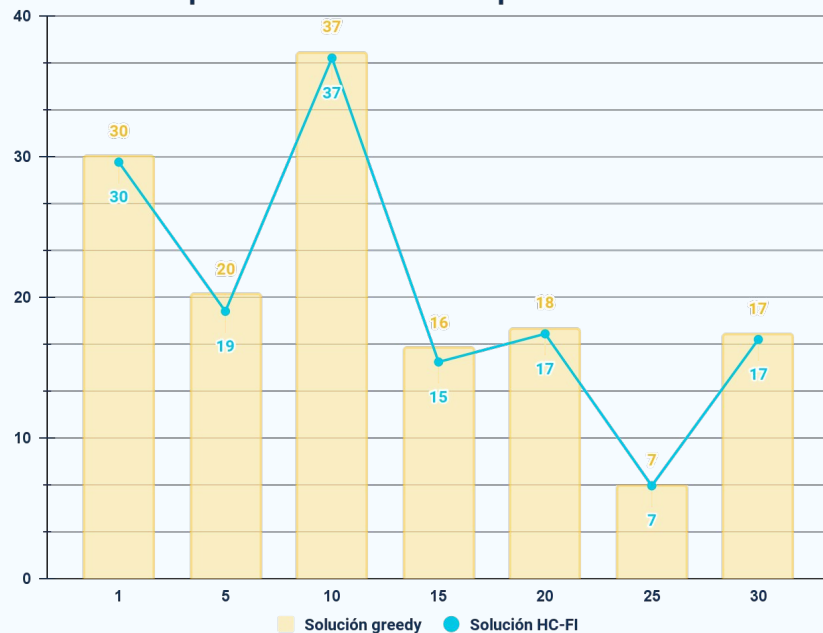


Resultados Instancias de Prueba

Alturas promedio encontradas para Instancias 2-5

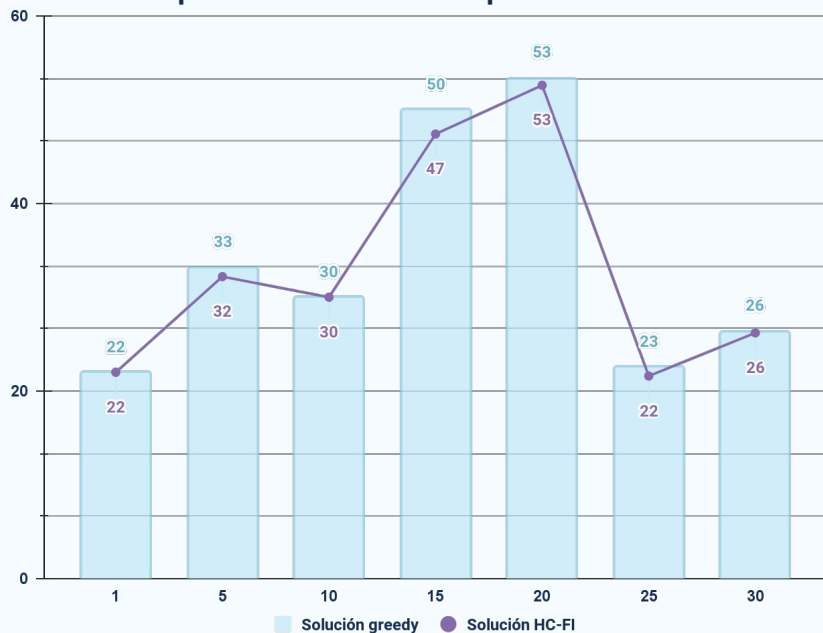


Alturas promedio encontradas para Instancias 6-10



Resultados Instancias de Prueba

Alturas promedio encontradas para Instancias 11-15

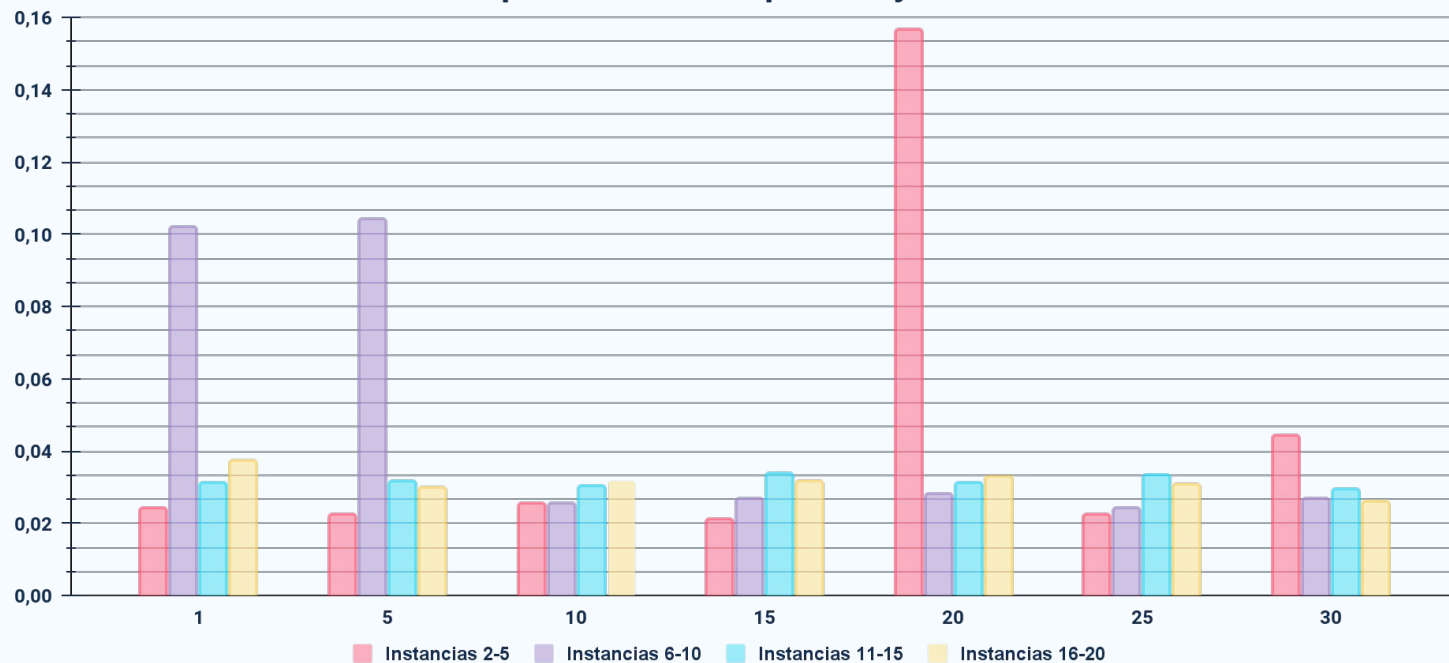


Alturas promedio encontradas para Instancias 16-20



Resultados Instancias de Prueba

Comparación de tiempos de ejecución



Conclusiones

