JUNIO

PL 1 – Planificación Automática

EXTRAORDINARIA 2024-25

JAVIER ROMA RODRIGUEZ

Table of Contents

Ejercicio 1: Planificación clásica con PDDL

Apartado 1.1: Logística de servicio de emergencias, versión inicial

Para realizar el Ejercicio he definido un dominio con los siguientes tipos:

```
(:types
        object
        dron caja person - object
        location
        content
    )
```

Y los siguientes predicados:

```
(:predicates
        (at ?o - object ?l - location)
        (holding-left  ?d - dron ?c - caja)
        (holding-right ?d - dron ?c - caja)
        (left-free     ?d - dron)
        (right-free    ?d - dron)

        ;;cajas
        (contenido-caja ?b - caja ?content - content)
        (person-needs ?p - person ?content - content)
        (person-has ?p - person ?content - content)
    )
```

He unificado las localizaciones de los drones, caja y personas heredando del tipo objeto, y definido las dos garras del dron(en el momento de definir el dominio lo hice pensando en usar varios drones por eso el ?d en los holding-(left/right)), luego (left/right)-free es para la disponibilidad de la garra, contenido caja para asociar tipos de contenido a las cajas, y person-(needs/has) para controlar que tiene y necesita una persona.

Apartado 1.2: Generador de problemas en Python

Para adaptar el generador a mi dominio he realizado los siguientes cambios:

- He añadido un parámetro Verbosity para facilitar el debug, y quitado el parámetro carriers ya que en este dominio no se usa.

```
parser.add_option('-v', '--verbosity', metavar='LEVEL', type=int, dest='verbosity',
                  default=1, help='nivel de verbosidad: 0=silencioso, 1=normal, 2=debug')
```

- Definido las Garras:

```
for x in drone:
    f.write(f"    (at {x} deposito)\n")
    f.write(f"    (left-free {x})\n")
    f.write(f"    (right-free {x})\n")
```

- Set de todas las cajas y personas en localizaciones

```
# Todas las cajas en deposito
for c in crate:
    loc = crate_locations[c]
    f.write(f"    (at {c} deposito)\n")

# Asociar cada caja con su contenido
for idx, crates in enumerate(crates_with_contents):
    for c in crates:
        f.write(f"    (contenido-caja {c} {content_types[idx]})\n")

# Todas las personas en ubicaciones aleatorias
for p in person:
    loc = person_locations[p]
    f.write(f"    (at {p} {loc})\n")
```

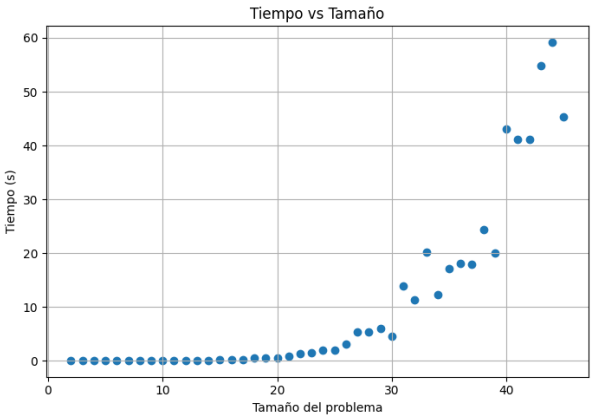
- Definición de las Metas:

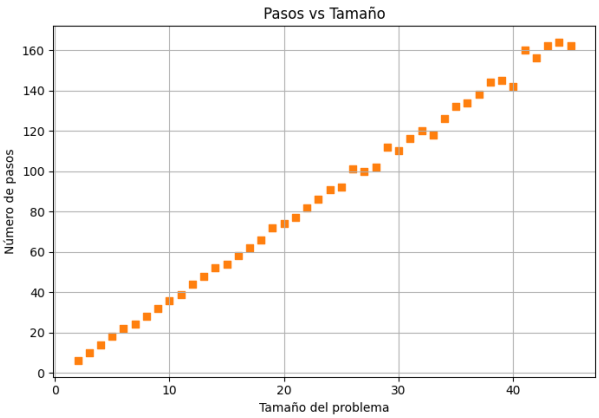
```
# Metas
f.write("  (:goal (and\n")
# Todos los drones deben terminar en el depósito
for x in drone:
    f.write(f"    (at {x} deposito)\n")
# Cada persona debe tener el contenido que necesita
for i in range(options.persons):
    for j in range(len(content_types)):
        if need[i][j]:
            p_name = person[i]
            c_type = content_types[j]
            f.write(f"    (person-has {p_name} {c_type})\n")

f.write("  ))\n")
```

Apartado 1.3: Comparativa rendimiento de planificadores

FF





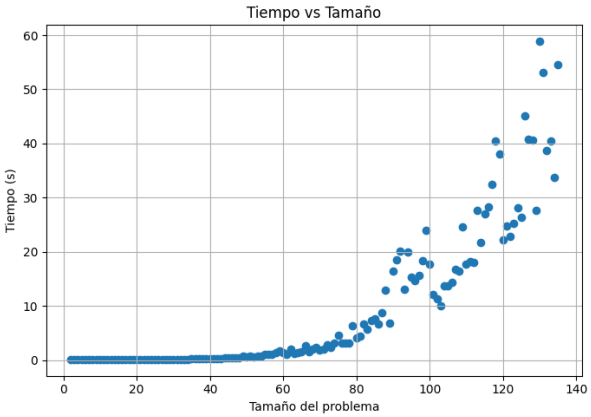
Métrica	Tamaño	Tiempo (s)	Pasos
Media	21.4	7.67	77.1
Mediana	21.0	1.18	74
Desviación Std	11.7	14.03	43.7
Mínimo	2	0	6
Máximo	44	59.02	164

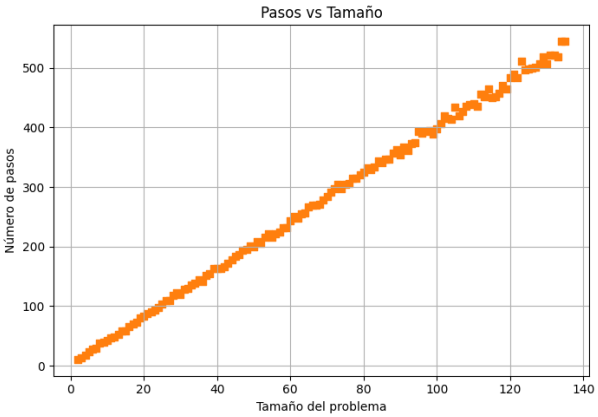
Ejecutando el planeador FF podemos observar que el tamaño máximo resuelto en menos de 1 minuto es de 44, según la gráfica podemos ver como el tiempo en resolver crece de manera exponencial.

- Para **problemas pequeños** (≤15–20 objetos), FF es casi instantáneo
- Para **problemas medianos** (20–35), observo subidas de tiempo, pero no demasiado grandes.
- Para **problemas grandes** (>35), el tiempo se dispara

Esto se debe al crecimiento del número de nodos a explorar y la profundidad y anchura del Árbol

LPG-TD



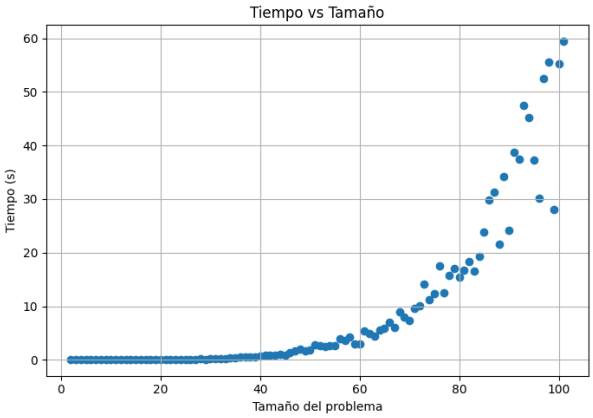


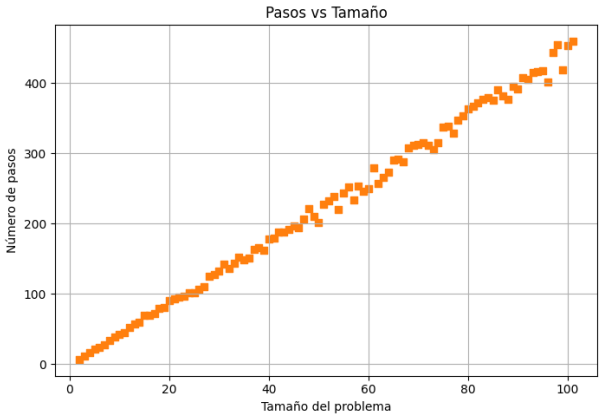
Estadísticas para LPG-TD

Métrica	Tamaño	Tiempo (s)	Pasos
Media	68.5	10.51	272.7
Mediana	68.5	2.50	270.5
Desviación Std	38.8	14.51	153.1
Mínimo	2.0	0.08	10.0
Máximo	135.0	58.93	545.0

Ejecutando el planeador LPG-TD podemos observar que el tamaño máximo resuelto en menos de 1 minuto es de 135, la comparativa de tamaño es inmensa, no solo calcula más planes en el tiempo determinado, sino que lo hace manteniendo una calidad similar en cuanto al número de pasos se refiere.

SGPLAN



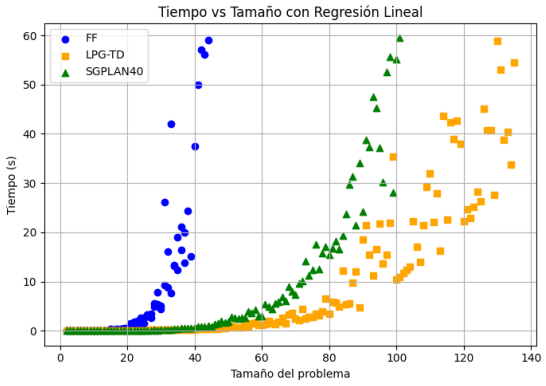


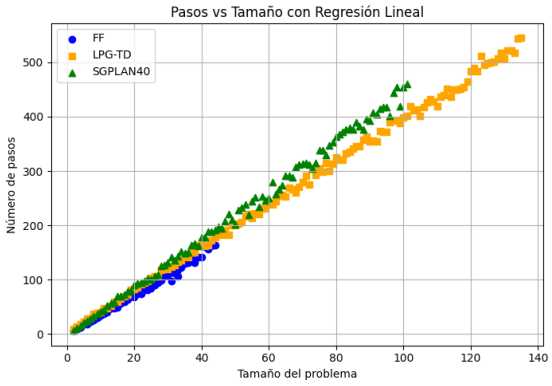
Estadísticas para SGPLAN40

Métrica	Tamaño	Tiempo (s)	Pasos
Media	51.5	9.71	226.0
Mediana	51.5	2.56	224.5
Desviación Std	29.0	14.78	129.9
Mínimo	2.0	0.00	6.0
Máximo	101.0	59.48	460.0

El planeador SGPLAN40 consigue también un número muy elevado de planes en 60 segundos, aunque podemos observar que la pendiente de la recta de Pasos-Tamaño mayor a las otras dos indicando que el planificador ofrece soluciones de peor calidad en cuanto a número de pasos se refiere

GRÁFICA COMPARATIVA Y CONCLUSIONES





En conclusión, observando las gráficas de arriba podemos determinar que el mejor planificador para este caso en específico (60 seg) en este problema es el LPG-D ya que ofrece una mayor rapidez respecto a los otros dos planificadores sin sacrificar Tamaño del plan.

Ejercicio 2: Planificación con números y optimización

Apartado 2.1: Servicio de emergencias, transportadores

Para cumplir con los requisitos del enunciado, he modificado el dominio realizando las siguientes acciones:

- He definido 5 acciones nuevas, poner y sacar caja del transportador, coger y dejar el transportador, y mover el transportador.
- He definido los números tanto en el generador como en el dominio.
- He definido predicado cajas-en(capacidad transportador), siguiente-t(gestión de números), llevado-por(d - dron lleva t - transportador) y en-transportador(c-caja está en t-transportador).
- He reducido el número de garras de los drones a 1.
- He adaptado la estructura del generador de problemas para los nuevos predicados.

Sin asignación de costes no se utilizan a no ser que elimines mover-dron para forzar el uso del transportador.

```
cargar_dron dron1 crate1 deposito (0)
poner-caja-en-transportador dron1 carrier1 crate1 deposito n0 n1 (0)
coger-transportador dron1 carrier1 deposito (0)
mover-transportador dron1 carrier1 deposito loc2 (98)
dejar-transportador dron1 carrier1 loc2 (0)
sacar-caja-del-transportador dron1 carrier1 crate1 loc2 n0 n1 (0)
entregar person1 crate1 comida loc2 dron1 (0)
```

El fragmento de arriba muestra el funcionamiento de los transportadores y el de abajo el de la numeración de la capacidad

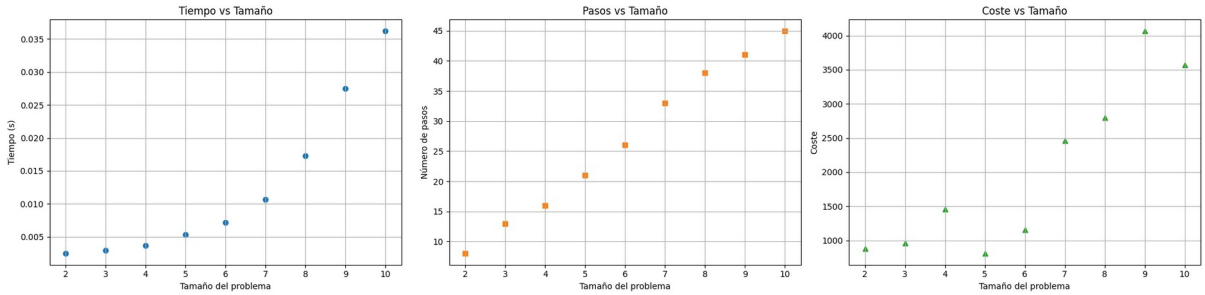
```
(cargar_dron dron1 crate1 deposito)
(poner-caja-en-transportador dron1 carrier1 crate1 deposito n0 n1)
(cargar_dron dron1 crate2 deposito)
(poner-caja-en-transportador dron1 carrier1 crate2 deposito n1 n2)
(cargar_dron dron1 crate3 deposito)
(poner-caja-en-transportador dron1 carrier1 crate3 deposito n2 n3)
(cargar_dron dron1 crate4 deposito)
(poner-caja-en-transportador dron1 carrier1 crate4 deposito n3 n4)
(coger-transportador dron1 carrier1 deposito)
(mover-transportador dron1 carrier1 deposito loc4)
(dejar-transportador dron1 carrier1 loc4)
(sacar-caja-del-transportador dron1 carrier1 crate1 loc4 n3 n4)
(entregar person3 crate1 comida loc4 dron1)
(sacar-caja-del-transportador dron1 carrier1 crate4 loc4 n2 n3)
```

Si probamos todos los planificadores el único que lo usa sin forzar(para un tamaño 5) es LPG-TD y no siempre, de hecho las únicas veces que le he visto usarlo ha sido sin utilidad alguna, como las acciones no conllevan un coste el planificador opta por usar

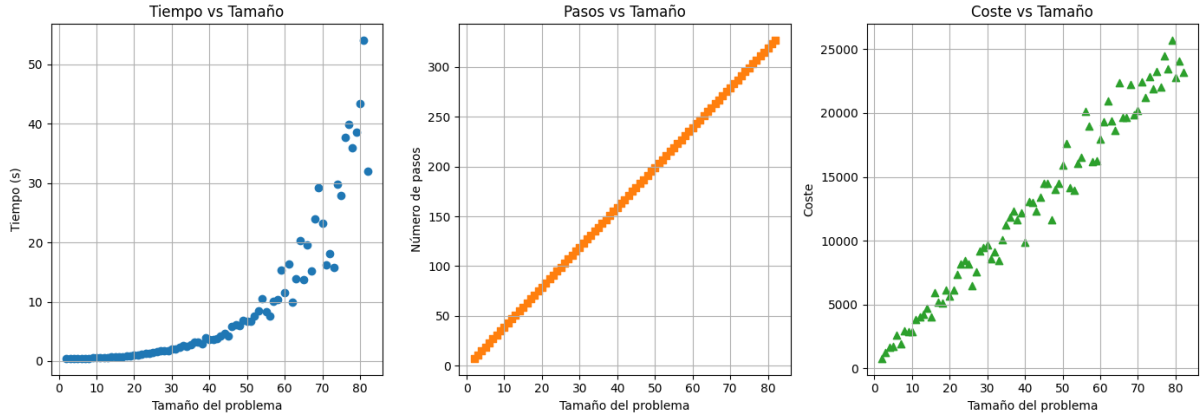
el dron, en el momento en el que eso cambia, sí que usan el transportador pero claro, otros planificadores que soportan costes

Apartado 2.2: Servicio de emergencias, costes de acción

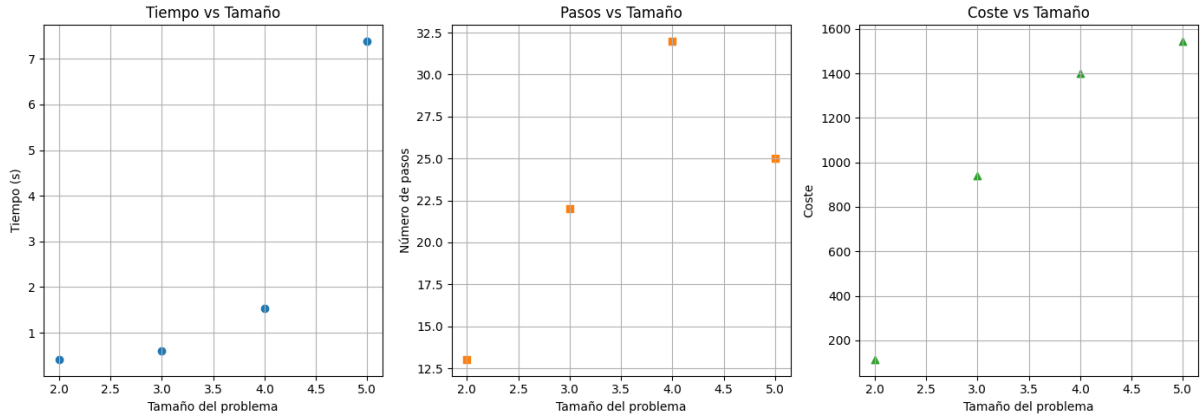
MetricFF



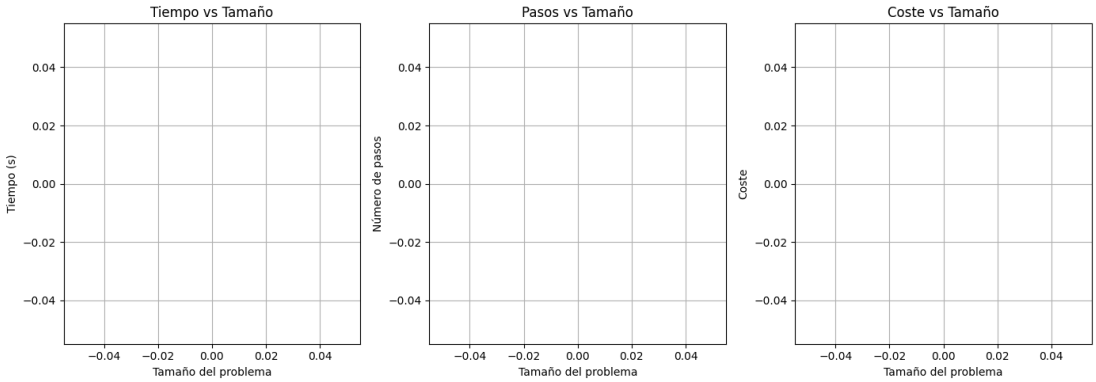
Lama First



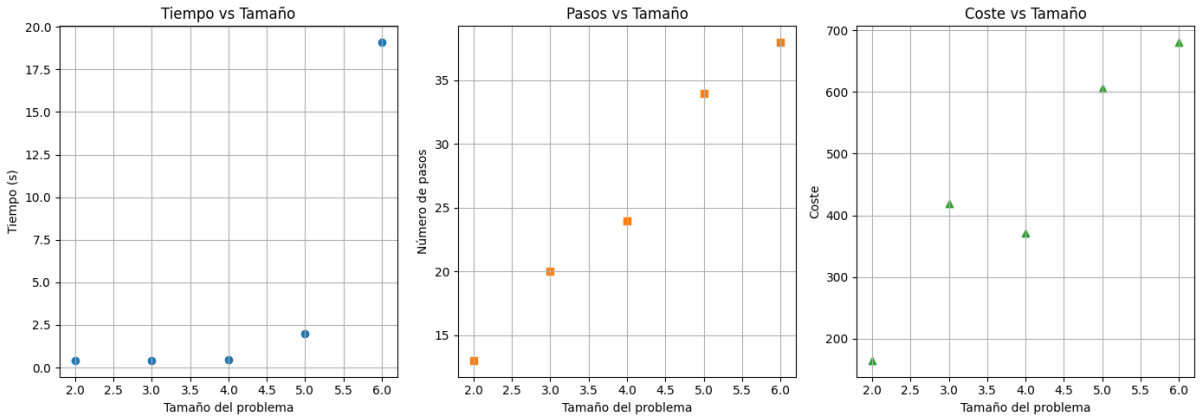
Autotune 2



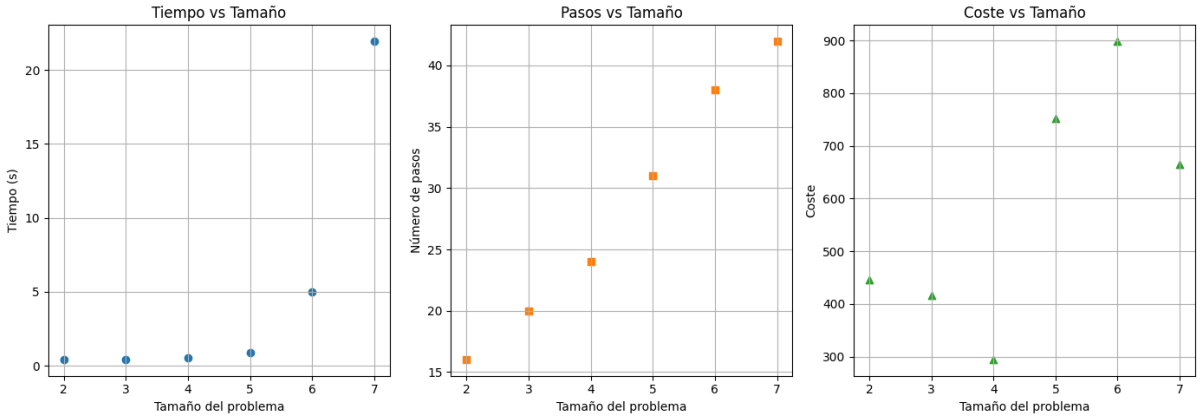
Stone Soup 2 Satisficing version



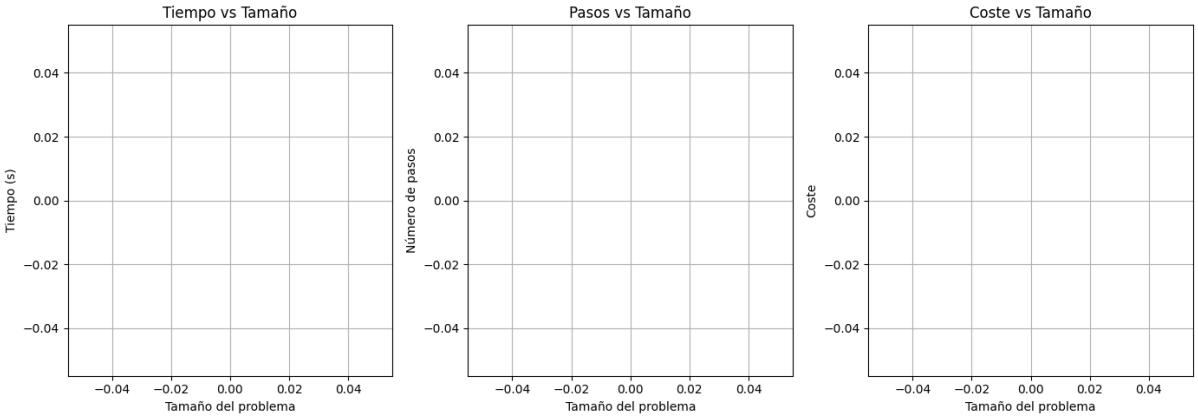
seq-opt-lmcut



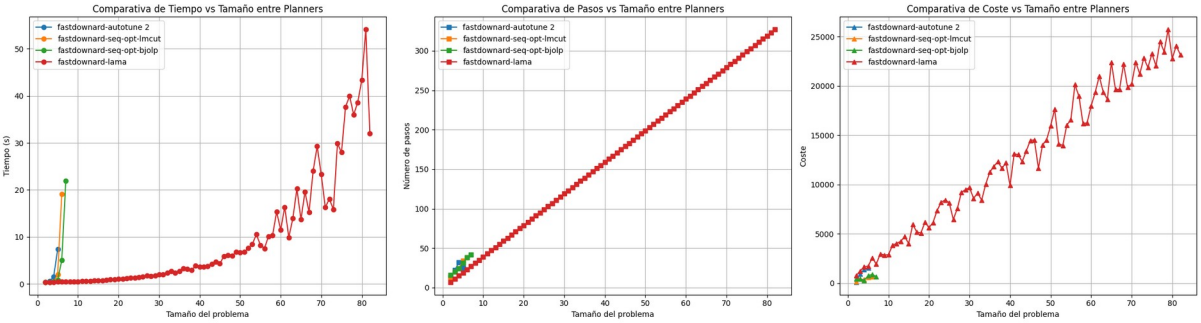
BJOLP



Stone Soup 2 Satisficing version



Gráfica Comparativa y Conclusiones



Observando los resultados, determino que el planificador que mejor funciona en este caso es fasdownard-lama que ofrece una cantidad muy superior de soluciones respecto a los otros planeadores que han encontrado solución en menos de 60 segundos.

Haciendo la prueba con tamaño 5, todos los planificadores que encuentran solución(los mismos que aparecen en la gráfica) lo hacen usando el transportador.

Planificador/Alias	Tamaño máximo resuelto	Coste solución	Tipo
fastdownward-autotune 2	5	1 546.0	porfolio
fastdownward-seq-opt-lmcut	6	681.0	óptimo
fastdownward-seq-opt-bjolp	7	665.0	óptimo
fastdownward-lama	82	23 168.0	anytime

Observando detenidamente las salidas, **bjolp** ofrece mejores planes en cuanto a coste/tamaño se refiere pero tarda bastante más en obtener un plan, ademas que lama lo hace en un numero menor de pasos

Ejercicio 3: Planificación con concurrencia

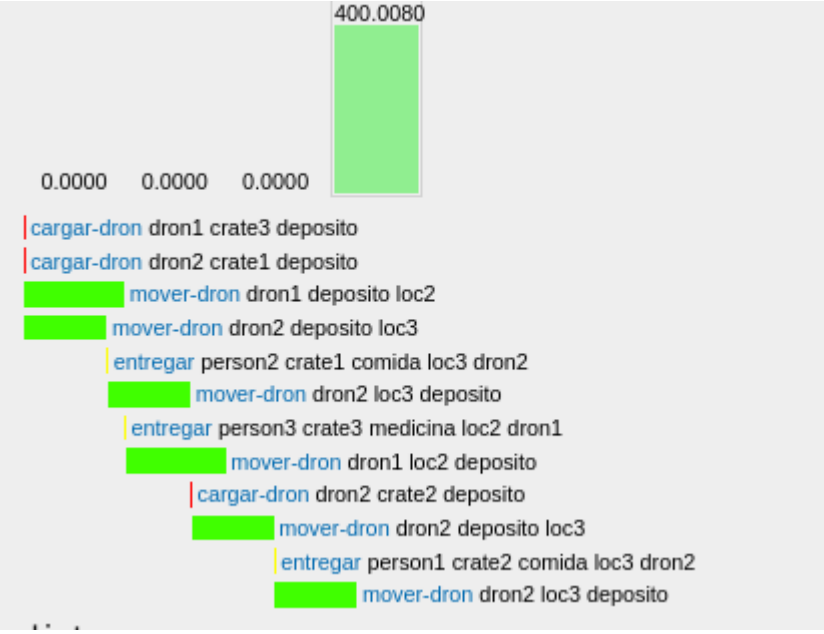
Apartado 3.1: Acciones concurrentes en gestión de emergencias

He implementado varios predicados a medida de Mutex, para evitar que dos drones carguen la misma caja(de una localización o de un transportador), que entreguen a la misma persona y que cojan el mismo transportador. el resto de acciones no hay problema de que las realicen a la vez.

Apartado 3.2: Implementación y pruebas de concurrencia

El tamaño máximo resuelto en menos de 60 seg ha sido 7 con 2 drones y 2 transportadores, todo esto deshabilitado la optimización de optic, sino tarda más debido a la optimización y no es capaz de resolver ninguno en menos de 1 minuto.

en esta salida para tamaño 8(sin optimizar) podemos observar que si es concurrente y hay varias acciones transcurriendo en la misma instancia de tiempo, si lo reducimos a tamaño 2, se ve más clara la concurrencia.



Si aumentamos el tamaño del problema en los parametros localizaciones y crates y personas, el tiempo de planificación crece exponencialmente. Sin embargo si lo hacemos con drones algunos problemas encuentran solucion más rapido

Por otra parte si obligamos a los drones a usar el transportador algunos problemas se resuelven en un tiempo menor. Este de abajo es un problema de 6 cajas que ha encotrado solucion más rapido con 3 drones que con 2, aunque ha tardado más en optimizarlo

Aclaración Final

Todas las respuestas se basan en la observación de los resultados obtenidos, en la carpeta del documento hay un jupyter notebook donde se muestran(en código y respectivos resultados) todos los resultados obtenidos así como la forma de obtenerlos. En principio se podrían ejecutar nada mas descargarse, quitando los planificadores temporales que requieren de la instalación de la biblioteca adicional.