UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI
INGEGNERIA INFORMATICA,
MODELLISTICA, ELETTRONICA
E SISTEMISTICA
DIMES

Master's Degree in Computer Engineering

Report

Course Assignment of Artificial Intelligence

Federica Cosenza (matr. 242445)
Michele Purrone (matr. 242457)
Antonino Vaccarella (matr. 250135)

Academic Year 2022/2023

Index

TASK 1

PDDL (Planning Domain Definition Language) is a planning domain description language for the definition and modelling of planning problems, created with the aim of acting as a standard to enable developers to communicate in a uniform and interoperable manner.

The team's focus, in the context of the project's first task, was the use of PDDL version 1.2 [1]. This version, based on the concepts already defined for STRIPS, was used in the AIPS 1998 competition. It offers a set of clear and well-defined constructs that allow the problem domain, possible actions and objectives to be specified.

In the remainder of this report, the use of PDDL 1.2 will be examined in detail, analysing domain modelling, implementation considerations and the results obtained.

DOMAIN DESCRIPTION

Consider a logistical problem of the emergency services, where a number of injured individuals are in fixed, non-moving positions. The purpose of planning is to organise the activities of one or more robotic agents in order to deliver boxes containing emergency supplies to each person. The assumptions are given in the outline.

The domain file (*domain.pddl*), contained in the Task 1 directory, includes:

Types

- **Location**: represents the location of a box, agent, content, carrier or person.
- **Agent:** represents the robotic agent.
- **Box:** represents the box.
- **Content:** represents the contents of the box.
- **Carrier:** represents the carrier.
- **Person:** represents the person.
- **Box_number:** represents the number of boxes.

Predicates

- **(at ?x - (either agent box content carrier person) ?l - location)** indicates the location l of an entity x.
- **(empty ?b - box)** indicates that box b is empty.
- **(filled ?b - box ?c - content)** indicates that box b has been filled with content c.
- **(carrier_loaded ?c - carrier ?b - box)** indicates that carrier c was loaded with box b.
- **(needs ?p - person ?c - content)** indicates that person p needs content c.
- **(has ?p - person ?c -** content**)** indicates that person p has obtained content c.
- **(capacity ?c - carrier ?bn - box_number)** indicates the current capacity of the carrier, i.e. the number of boxes transported.
- **(next ?c - carrier ?bn1 ?bn2 - box_number)** defines an ordering between the possible values of box_number, establishing which values are next and previous.

Actions

1. **fill_box**: This action has the task of filling an empty box with a certain content. Its parameters are ?*robot* (the agent performing the action), *?*box (the box to be filled), *?*content (the content to be placed in the box) and *?at* (the position of the agent and the box). The precondition requires that the agent, the box and the content are in the same position and that the box is empty. The effect of the action is to fill the box with the indicated contents.

```
(:action fill_box
  :parameters (?robot - agent ?box - box ?content - content ?at - location)
  :precondition (and (at ?robot ?at) (at ?box ?at) (empty ?box) (at ?content ?at))
  :effect (and (not (empty ?box)) (filled ?box ?content))
)
```

2. **load_box_on_carrier**: this action deals with loading a box onto a carrier. Its parameters are ?*robot*, ?*box, ?content*, *?carrier* (the conveyor on which to load the box), ?*at* (the position of the agent, box and conveyor) and ?*bnbefore* and ?*bnafter* (used to manipulate the capacity of the conveyor before and after loading the box). The precondition requires that the agent, box, contents and conveyor are in the same position, that the box is full and that the conveyor still has space available to load the box. The effect of the action is to load the box onto the conveyor, thus changing its current capacity.

```
(:action load_box_on_carrier
  :parameters (?robot - agent ?box - box ?content - content ?carrier - carrier ?at - location ?bnbefore ?bnafter - box_number)
  :precondition (and (at ?robot ?at) (at ?box ?at) (filled ?box ?content) (at ?carrier ?at) (capacity ?carrier ?bnbefore)
                (next ?carrier ?bnbefore ?bnafter))
  :effect (and (not (at ?box ?at)) (carrier_loaded ?carrier ?box) (not (capacity ?carrier ?bnbefore)) (capacity ?carrier ?bnafter))
)
```

3. **unload_box_from_carrier**: this action has the task of unloading a box from a carrier. Its parameters are ?*robot*, ?*box*, ?*carrier*, ?*at* and ?*bnbefore* and ?*bnafter*. The precondition requires that the agent, box and carrier are in the same position and that the box is empty. The effect of the action is to unload the box from the carrier, thus changing its current capacity.

```
(:action unload_box_from_carrier
  :parameters (?robot - agent ?box - box ?carrier - carrier ?at - location ?bnbefore ?bnafter - box_number)
  :precondition (and (at ?robot ?at) (carrier_loaded ?carrier ?box)  (empty ?box) (at ?carrier ?at) (capacity ?carrier ?bnbefore)
                (next ?carrier ?bnafter ?bnbefore))
  :effect (and (at ?box ?at) (not (carrier_loaded ?carrier ?box))  (not (capacity ?carrier ?bnbefore)) (capacity ?carrier ?bnafter))
)
```

4. **move_with_carrier**: this action is responsible for moving the agent and the carrier from one position to another. Its parameters are ?*robot*, ?*carrier,* ?*from* (the start position), ?*to* (the end position). The precondition requires the agent and the means of transport to be at the start position. The effect of the action is to move the agent and the means of transport to the target position.

```
(:action move_with_carrier
  :parameters (?robot - agent ?carrier - carrier ?from - location ?to - location)
  :precondition (and (at ?robot ?from) (at ?carrier ?from))
  :effect (and
            (not (at ?robot ?from))
            (not (at ?carrier ?from))
            (at ?robot ?to)
            (at ?carrier ?to))
)
```

5. **deliver_content**: this action has the task of delivering the contents of a box to a person who needs it. Its parameters are ?*robot*, ?*box*, *?content, ?carrier*, ?*person* (the person to whom the content is to be delivered) and *?at* (the location of the agent, the box, the carrier and the person). The precondition requires that the agent, the box, the contents, the carrier and the person are in the same position and that the box is full. The effect of the action is to deliver the contents of the box to the person indicated by emptying the box.

```
(:action deliver_content
  :parameters (?robot - agent ?box - box ?content - content ?carrier - carrier ?person - person ?at - location)
  :precondition (and (at ?robot ?at) (filled ?box ?content) (at ?carrier ?at) (carrier_loaded ?carrier ?box)  (needs ?person ?content) (at ?person ?at))
  :effect (and (empty ?box) (not (filled ?box ?content)) (has ?person ?content) (at ?person ?at) (at ?robot ?at) (at ?carrier ?at))
)

)
```

DESIGN CHOICES

Given the impossibility of using *fluents* and *functions* to represent numerical values, an alternative solution was chosen, using only constructs from PDDL version 1.2. In order to count how many boxes were on the cart and keep track of the cart's capacity, the predicate **next** was used. This allowed the **capacity** value to be increased. For example, in the first problem file, capacity has *zero*, *one*, *two, three* and *four as* permissible values*:* this prevents capacity from exceeding the maximum limit imposed by the trace.

```
(next carrier zero one)
(next carrier one two)
(next carrier two three)
(next carrier three four)
```

The capacity value is increased as an effect of the *load_box_on_carrier action* and decreased after the *unload_box_from_carrier* action.

The assumption was made that the robot only moves with the trolley. The action describing the movement of the agent (**move**) was commented out as it is never used by the robot, which rather moves with the trolley to transport the contents needed by the injured.

TASK 2

Inside the Task 2 directory, there is the Problems folder, which includes the three instances of the problems required by the trace, called *problem1*, *problem2* and *problem3* respectively.

problem1.pddl > ...

```
(define (problem emergency_aid)
  (:domain emergency_services)
  no commands
  (:objects
    depot loc2 loc3 - location
    robot - agent
    box1 box2 box3 box4 box5 - box
    food medicine - content
    carrier - carrier
    zero one two three four - box_number
    p1 p2 p3 - person
  )
  no commands
  (:init
    (at robot depot)
    (at box1 depot)
    (at box2 depot)
    (at box3 depot)
    (at box4 depot)
    (at box5 depot)
    (empty box1)
    (empty box2)
    (empty box3)
    (empty box4)
    (empty box5)
    (at food depot)
    (at medicine depot)
    (at carrier depot)
    (capacity carrier zero)
    (next carrier zero one)
    (next carrier one two)
    (next carrier two three)
    (next carrier three four)
    (at p1 loc2)
    (at p2 loc2)
    (at p3 loc3)
    (needs p1 food)
    (needs p1 medicine)
    (needs p2 medicine)
    (needs p3 food)

  )
  (:goal
    (and
      (has p1 food)
      (has p1 medicine)
      (has p2 medicine)
      (has p3 food)
    )
  )
)
```

Figure 1: problem1.pddl file

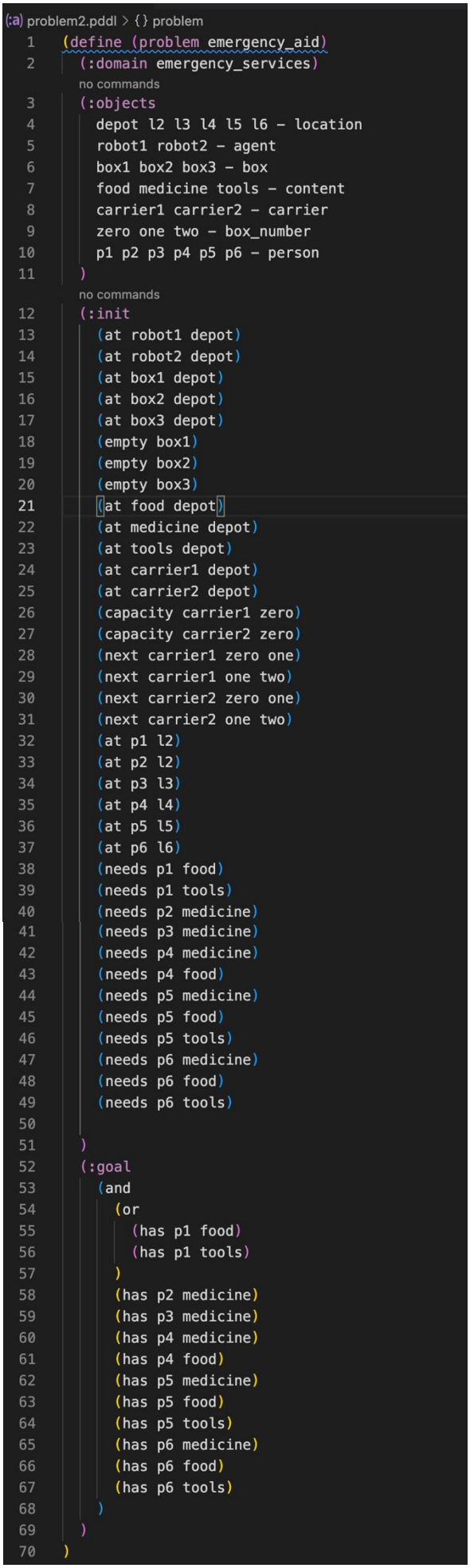

Figure 2: problem2.pddl file

problem3.pddl > ...

```
(define (problem emergency_aid)
  (:domain emergency_services)
  no commands
  (:objects
    depot l2 l3 l4 l5 l6 l7 l8 - location
    robot1 robot2 - agent
    box1 box2 box3 box4 - box
    food medicine tools - content
    carrier1 carrier2 - carrier
    zero one two - box_number
    p1 p2 p3 p4 p5 p6 p7 p8 - person
  )
  no commands
  (:init
    (at robot1 depot)
    (at robot2 depot)
    (at box1 depot)
    (at box2 depot)
    (at box3 depot)
    (at box4 depot)
    (empty box1)
    (empty box2)
    (empty box3)
    (empty box4)
    (at food depot)
    (at medicine depot)
    (at tools depot)
    (at carrier1 depot)
    (at carrier2 depot)
    (capacity carrier1 zero)
    (capacity carrier2 zero)
    (next carrier1 zero one)
    (next carrier1 one two)
    (next carrier2 zero one)
    (next carrier2 one two)
    (at p1 l2)
    (at p2 l2)
    (at p3 l3)
    (at p4 l4)
    (at p5 l5)
    (at p6 l6)
    (at p7 l7)
    (at p8 l8)
    (needs p1 food)
    (needs p1 tools)
    (needs p2 medicine)
    (needs p3 medicine)
    (needs p4 medicine)
    (needs p4 food)
    (needs p5 medicine)
    (needs p5 food)
    (needs p5 tools)
    (needs p6 medicine)
    (needs p6 food)
    (needs p6 tools)
    (needs p7 medicine)
    (needs p7 food)
    (needs p7 tools)
    (needs p8 medicine)
    (needs p8 food)
    (needs p8 tools)

  )
  (:goal
    (and
      (or
        (has p1 food)
        (has p1 tools)
      )
      (has p2 medicine)
      (has p3 medicine)
      (has p4 medicine)
      (has p4 food)
      (has p5 medicine)
      (has p5 food)
      (has p5 tools)
      (has p6 medicine)
      (has p6 food)
      (has p6 tools)
      (has p7 medicine)
      (has p7 food)
      (has p7 tools)
      (has p8 medicine)
      (has p8 food)
      (has p8 tools)
    )
  )
)
```

Figure 3: problem3.pddl file

When drafting the problems, the planning requirements of the track were taken into account, so the 'or' construct was included in the goal section. However, since both PDDL4J and the adopted planner (Pyperplan) are unable to handle the disjunction between predicates, it was decided to relax the problem and put all the goals of problems two and three in and. For the sake of completeness, however, a plan generated by PDDL Editor is shown to show support for the “and” and generation of a plan taking into account the requirements of the original problem.

Found Plan (output)

(fill_box robot1 box3 medicine depot)
(fill_box robot1 box2 food depot)
(load_box_on_carrier robot1 box3 medicine carrier2 depot zero one)
(load_box_on_carrier robot1 box2 food carrier2 depot one two)
(move_with_carrier robot1 carrier2 depot l6)
(fill_box robot2 box1 tools depot)
(load_box_on_carrier robot2 box1 tools carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l5)
(deliver_content robot2 box1 tools carrier1 p5 l5)
(move_with_carrier robot2 carrier1 l5 depot)
(unload_box_from_carrier robot2 box1 carrier1 depot one zero)
(fill_box robot2 box1 tools depot)
(load_box_on_carrier robot2 box1 tools carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l2)
(deliver_content robot2 box1 tools carrier1 p1 l2)
(move_with_carrier robot2 carrier1 l2 depot)
(unload_box_from_carrier robot2 box1 carrier1 depot one zero)
(fill_box robot2 box1 tools depot)
(load_box_on_carrier robot2 box1 tools carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l6)
(deliver_content robot2 box1 tools carrier1 p6 l6)
(move_with_carrier robot1 carrier1 l6 l4)
(unload_box_from_carrier robot1 box1 carrier1 l4 one zero)
(move_with_carrier robot2 carrier2 l6 l4)
(move_with_carrier robot1 carrier1 l4 l6)
(deliver_content robot2 box2 food carrier2 p4 l4)
(move_with_carrier robot2 carrier2 l4 depot)
(unload_box_from_carrier robot2 box2 carrier2 depot two one)
(fill_box robot2 box2 food depot)
(load_box_on_carrier robot2 box2 food carrier2 depot one two)
(move_with_carrier robot2 carrier2 depot l6)
(deliver_content robot2 box2 food carrier2 p6 l6)
(move_with_carrier robot2 carrier2 l6 l5)
(move_with_carrier robot1 carrier1 l6 depot)
(deliver_content robot2 box3 medicine carrier2 p5 l5)
(move_with_carrier robot2 carrier2 l5 depot)
(move_with_carrier robot2 carrier1 depot l6)
(unload_box_from_carrier robot1 box3 carrier2 depot two one)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)

(fill_box robot1 box2 medicine depot)

(load_box_on_carrier robot1 box2 medicine carrier2 depot zero one)

(fill_box robot1 box3 food depot)

(load_box_on_carrier robot1 box3 food carrier2 depot one two)

(move_with_carrier robot1 carrier2 depot l5)

(deliver_content robot1 box3 food carrier2 p5 l5)

(move_with_carrier robot1 carrier2 l5 l4)

(deliver_content robot1 box2 medicine carrier2 p4 l4)

(move_with_carrier robot1 carrier2 l4 depot)

(unload_box_from_carrier robot1 box3 carrier2 depot two one)

(fill_box robot1 box3 medicine depot)

(move_with_carrier robot1 carrier2 depot l2)

(move_with_carrier robot2 carrier1 l6 depot)

(load_box_on_carrier robot2 box3 medicine carrier1 depot zero one)

(move_with_carrier robot2 carrier1 depot l2)

(move_with_carrier robot1 carrier1 l2 l3)

(move_with_carrier robot2 carrier2 l2 l6)

(deliver_content robot1 box3 medicine carrier1 p3 l3)

(move_with_carrier robot1 carrier1 l3 depot)

(unload_box_from_carrier robot1 box3 carrier1 depot one zero)

(move_with_carrier robot2 carrier2 l6 depot)

(move_with_carrier robot2 carrier1 depot l6)

(unload_box_from_carrier robot1 box2 carrier2 depot one zero)

(fill_box robot1 box2 medicine depot)

(load_box_on_carrier robot1 box2 medicine carrier2 depot zero one)

(move_with_carrier robot1 carrier2 depot l2)

(deliver_content robot1 box2 medicine carrier2 p2 l2)

(move_with_carrier robot1 carrier2 l2 depot)

(unload_box_from_carrier robot1 box2 carrier2 depot one zero)

(fill_box robot1 box3 medicine depot)

(load_box_on_carrier robot1 box3 medicine carrier2 depot zero one)

(move_with_carrier robot1 carrier2 depot l6)

(deliver_content robot2 box3 medicine carrier2 p6 l6)

(reach-goal)

Figure 4: Plan generated by PDDL Editor with problem2.pddl

As previously mentioned, for the realisation of this task, the Pyperplan planner [2], a lightweight STRIPS planner written in Python, was used. This planner includes a wide range of search algorithms, including:

- Breadth-first search
- Enforced hill-climbing search
- (Weighted) A* search
- Greedy best-first search
- Iterative deepening search
- SAT planner

As far as heuristics are concerned, Pyperplan provides the following options:

- Blind
- hAdd
- Set-additive
- hMax
- Landmark-cut heuristic
- Landmark heuristic

A notable aspect of Pyperplan is its highly extensible and modifiable nature, allowing the development of an additional search algorithm and other heuristics not present in the original planner source code.

This planner is contained in the folder *Task2/pyperplain-main*. Within this folder, under the path *pyperplan-main\pyperplan\search\astar.py you will find the* search algorithm and within *pyperplan-main\pyperplan\heuristics\relaxation.py you will* find the implemented heuristics.

SEARCH ALGORITHM

The search algorithm implemented is ***Memory Bounded A**** (MBA*).

The use of this algorithm was advantageous given the complexity of the problems and the large number of nodes generated. This algorithm inherits all the features of A*, which is already present within the planner, but with a focus on memory.

An *open* heap is used to keep the nodes still to be explored, sorted according to the estimated total cost (f) to reach the goal. A main loop is executed that continues as long as there are nodes in *open*. During each iteration, the node with the lowest estimated total cost is extracted, its successors are explored and added into *open if they have* not yet been visited or if a cheaper way to reach them has been found. At each iteration, an integer *counter* parameter is incremented to count the number of values placed in the queue. Every so many values (in this case a thousand), a memory check takes place: if the size of the queue exceeds the maximum memory limit (*max_memory*), the queue is "pruned", keeping only the smallest nodes. The advantage is obvious: reducing the amount of memory required to maintain the priority queue.

The modified section of A* that allows the memory size to be controlled is as follows:

```
#CONTROLLO MEMORIA
if counter % 1000 == 0: #ogni mille nodi
        if len(open) > max_memory: #se la lunghezza di open supera la soglia massima
            logging.info("Memory limit reached. Pruning open nodes.")
            # utilizza la funzione nsmallest() del modulo heapq per ottenere i primi max_memory elementi più piccoli da
            # open e assegna il risultato a open.

            open = heapq.nsmallest(max_memory, open)
```

The annotated algorithm code follows.

```python
def mb_astar_search(task, heuristic, make_open_entry=ordered_node_astar, use_relaxed_plan=False):
    max_memory = 1000 #limite di memoria
    open = []        #coda di priorità
    state_cost = {task.initial_state: 0}  #state_cost tiene traccia del costo g accumulato per raggiungere uno stato specifico durante la ricerca.
    node_tiebreaker = 0

    #seleziona il nodo radice e calcola il suo valore euristico
    root = searchspace.make_root_node(task.initial_state)
    init_h = heuristic(root)

    #il nodo radice viene inserito nell'heap open (l'heap garantisce che il nodo più piccolo sia sempre posizionato nella posizione radice).
    #Il nodo lo seleziona mediante il metodo ordered_node_astar.
    #messo a disposizione dalla classe, il quale preleva il nodo con il valore di h più piccolo
    heapq.heappush(open, make_open_entry(root, init_h, node_tiebreaker))
    logging.info("Initial h value: %f" % init_h)

    besth = float("inf")  #viene inizializzato a + infinito un parametro che indica il valore della migliore euristica
    counter = 0         #conta il numero di nodi in open
    expansions = 0      #conta il numero di nodi espansi

    while open:         #finchè ci sono elementi in open
        (f, h, _tie, pop_node) = heapq.heappop(open) #PRELEVA DA open il nodo radice,
                                                     #chiamato pop_node e i suoi valori vengono assegnati a f, h, _tie e pop_node.
        if h < besth: #se il valore di h è minore di besth
            besth = h #viene assegnato un nuovo valore per l'euristica
            logging.debug("Found new best h: %d after %d expansions" % (besth, counter))

        pop_state = pop_node.state  #pop_state = stato del nodo prelevato (pop_node)
        #se il costo per raggiungere quello stato è uguale al costo g di pop_node
        #espandiamo questo nodo se il suo costo associato (g) è il costo più basso per raggiungere questo stato (cioè state_cost[pop_state])
        if state_cost[pop_state] == pop_node.g:
            expansions += 1
            if task.goal_reached(pop_state):
                logging.info("Goal reached. Start extraction of solution.")
                logging.info("%d Nodes expanded" % expansions)
                return pop_node.extract_solution()

            rplan = None
            #non usiamo un piano rilassato
            if use_relaxed_plan:
                (rh, rplan) = heuristic.calc_h_with_plan(searchspace.make_root_node(pop_state))
                logging.debug("relaxed plan %s " % rplan)

            #get_success_states restituisce una lista di coppie (op, new_state) dove "op" è l'operatore applicabile
            #e new_state è lo stato risultante quando op viene applicato a state
            for op, succ_state in task.get_successor_states(pop_state):
                if use_relaxed_plan:
                    if rplan and not op.name in rplan:
                        # Ignore this operator if it's not in the relaxed plan
                        logging.debug(
                            "removing operator %s << not a preferred operator" % op.name
                        )
                        continue
                    else:
                        logging.debug("keeping operator %s" % op.name)

                #costruisce un nuovo nodo successore del nodo corrente
                succ_node = searchspace.make_child_node(pop_node, op, succ_state)
                h = heuristic(succ_node)  #calcola il valore euristico del nodo successivo
                if h == float("inf"): #si ignorano gli stati che non  possono raggiungere il goal
                    continue
                #cerca il valore associato a succ_state in state_cost. se succ_state è presente, old_succ_g viene impostato con il suo valore.
                #altrimenti viene impostato con un valore infinito.
                old_succ_g = state_cost.get(succ_state, float("inf"))
                if succ_node.g < old_succ_g: #se viene trovato un percorso più economico per raggiungere succ_state
                    #incrementa una variabile node_tiebreaker per garantire un ordine stabile dei nodi nella coda prioritaria in caso di parità di priorità.
                    node_tiebreaker += 1
                    # inserisce il nodo successore succ_node nella coda prioritaria open. L'ordine di inserimento viene determinato da una combinazione di fattori:
                    #la valutazione euristica h del nodo,il costo del percorso succ_node.g e il valore di node_tiebreaker per risolvere eventuali parità.
                    heapq.heappush(open, make_open_entry(succ_node, h, node_tiebreaker))
                    state_cost[succ_state] = succ_node.g # aggiorna il costo associato a succ_state nel dizionario state_cost con il nuovo costo succ_node.g.


        counter += 1

    #CONTROLLO MEMORIA
    if counter % 1000 == 0: #ogni mille nodi
            if len(open) > max_memory: #se la lunghezza di open supera la soglia massima
                logging.info("Memory limit reached. Pruning open nodes.")
                # utilizza la funzione nsmallest() del modulo heapq per ottenere i primi max_memory elementi più piccoli da
                # open e assegna il risultato a open.

                open = heapq.nsmallest(max_memory, open)

    logging.info("No operators left. Task unsolvable.")
    logging.info("%d Nodes expanded" % expansions)
    return None
```

Figure 5: MBA Search Algorithm*

EURISTICS

RELAXED GOAL COUNT (RGC)

The first heuristic implemented is called **Relaxed Goal Count (RGC)** and estimates the number of goals that have not yet been achieved in the current state. The RGC heuristic does not consider distances or costs to reach the goal, preferring to provide a simple estimate based on the remaining goal count.

Heuristics are implemented by means of a class containing two methods: __init__ and __call__ . The former is a simple constructor while the latter contains the code that implements the actual heuristics.

In __call__, the list of remaining goals is calculated by comparing the facts in the state of the goal with the facts in the state of the current node. Finally, the heuristic h_value corresponds to the number of remaining goals.

```python
class hRGCHeuristic(_RelaxationHeuristic):

    """
    L'euristica RGC stima il numero di goal che non sono ancora stati raggiunti nello stato corrente.
    """

    def __init__(self, task): #Inizializzazione
        super().__init__(task)

    def __call__(self, node):
        remaining_goals = [fact for fact in self.goals if fact not in node.state] #Si va a verificare quali goal non sono stati ancora raggiunti.
        h_value = len(remaining_goals) #Il valore restituito è la cardinalità dell'insieme dei goal non ancora raggiunti.
        return h_value
```

Figure 6: hRGCHeuristic class (found in the relaxation.py file)

To test the efficiency of the Memory Bounded A* algorithm, it was decided to generate (using the same heuristic) the plan on the same problem, first with A* and then with Memory Bounded A*.

The Tracemalloc module was used to print out the memory used; the first value displayed refers to the current memory and the second to the peak memory value (corresponding to the maximum memory usage since the algorithm was invoked at the end of its execution).

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H hrgc -s mba domain.pddl problem1.pddl

2023-07-18 15:37:38,561 INFO     using search: mb_astar_search
2023-07-18 15:37:38,561 INFO     using heuristic: hRGCHeuristic
2023-07-18 15:37:38,561 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:37:38,563 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem1.pddl
2023-07-18 15:37:38,564 INFO     8 Predicates parsed
2023-07-18 15:37:38,564 INFO     5 Actions parsed
2023-07-18 15:37:38,564 INFO     20 Objects parsed
2023-07-18 15:37:38,564 INFO     0 Constants parsed
2023-07-18 15:37:38,564 INFO     Grounding start: emergency_aid
2023-07-18 15:37:38,569 INFO     Relevance analysis removed 0 facts
2023-07-18 15:37:38,569 INFO     Grounding end: emergency_aid
2023-07-18 15:37:38,569 INFO     65 Variables created
2023-07-18 15:37:38,569 INFO     276 Operators created
2023-07-18 15:37:38,570 INFO     Search start: emergency_aid
2023-07-18 15:37:38,570 INFO     Initial h value: 4.000000
2023-07-18 15:37:38,636 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:37:38,695 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:37:38,756 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:37:39,367 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:37:39,435 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:37:39,440 INFO     Goal reached. Start extraction of solution.
2023-07-18 15:37:39,440 INFO     12970 Nodes expanded
2023-07-18 15:37:39,458 INFO     Search end: emergency_aid
2023-07-18 15:37:39,459 INFO     Search time: 0.89
2023-07-18 15:37:39,459 INFO     Memory used: 0.1895 MB 43.28 MB
2023-07-18 15:37:39,459 INFO     Plan length: 17
```

Figure 7: MBA+ RGC execution of problem 1

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H hrgc -s astar domain.pddl problem1.pddl
2023-07-18 15:40:17,611 INFO     using search: astar_search
2023-07-18 15:40:17,611 INFO     using heuristic: hRGCHeuristic
2023-07-18 15:40:17,612 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:40:17,612 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem1.pddl
2023-07-18 15:40:17,613 INFO     8 Predicates parsed
2023-07-18 15:40:17,613 INFO     5 Actions parsed
2023-07-18 15:40:17,613 INFO     20 Objects parsed
2023-07-18 15:40:17,613 INFO     0 Constants parsed
2023-07-18 15:40:17,613 INFO     Grounding start: emergency_aid
2023-07-18 15:40:17,617 INFO     Relevance analysis removed 0 facts
2023-07-18 15:40:17,617 INFO     Grounding end: emergency_aid
2023-07-18 15:40:17,618 INFO     65 Variables created
2023-07-18 15:40:17,618 INFO     276 Operators created
2023-07-18 15:40:17,618 INFO     Search start: emergency_aid
2023-07-18 15:40:17,618 INFO     Initial h value: 4.000000
2023-07-18 15:40:24,464 INFO     Goal reached. Start extraction of solution.
2023-07-18 15:40:24,464 INFO     113028 Nodes expanded
2023-07-18 15:40:24,755 INFO     Search end: emergency_aid
2023-07-18 15:40:24,755 INFO     Search time: 7.1
2023-07-18 15:40:24,755 INFO     Memory used: 0.1389 MB 273.4 MB
2023-07-18 15:40:24,757 INFO     Plan length: 14
```

Figure 8: Execution with A + RGC of problem 1*

As can be seen, the amount of memory used by the two algorithms changes considerably (43.28 MB of Memory Bounded A* versus 273.4 MB of A*), as do the execution times (0.89 seconds of Memory Bounded A* versus 7.1 seconds of A*).

Despite the advantages of MBA*, however, A* shows itself capable of finding a shorter plan.

problem1.pddl.soln

```
1   (fill_box robot box4 medicine depot)
2   (fill_box robot box1 medicine depot)
3   (fill_box robot box2 food depot)
4   (fill_box robot box5 food depot)
5   (load_box_on_carrier robot box4 medicine carrier depot zero one)
6   (load_box_on_carrier robot box1 medicine carrier depot one two)
7   (load_box_on_carrier robot box2 food carrier depot two three)
8   (load_box_on_carrier robot box5 food carrier depot three four)
9   (move_with_carrier robot carrier depot loc2)
10  (deliver_content robot box4 medicine carrier p2 loc2)
11  (deliver_content robot box1 medicine carrier p1 loc2)
12  (deliver_content robot box2 food carrier p1 loc2)
13  (move_with_carrier robot carrier loc2 loc3)
14  (deliver_content robot box5 food carrier p3 loc3)
```

Figure 9: plane generated by A + RGC*

problem1.pddl.soln

```
1   (fill_box robot box4 medicine depot)
2   (load_box_on_carrier robot box4 medicine carrier depot zero one)
3   (move_with_carrier robot carrier depot loc2)
4   (deliver_content robot box4 medicine carrier p2 loc2)
5   (move_with_carrier robot carrier loc2 depot)
6   (fill_box robot box3 medicine depot)
7   (fill_box robot box5 medicine depot)
8   (fill_box robot box1 food depot)
9   (fill_box robot box2 food depot)
10  (load_box_on_carrier robot box5 medicine carrier depot one two)
11  (load_box_on_carrier robot box1 food carrier depot two three)
12  (load_box_on_carrier robot box2 food carrier depot three four)
13  (move_with_carrier robot carrier depot loc2)
14  (deliver_content robot box5 medicine carrier p1 loc2)
15  (deliver_content robot box1 food carrier p1 loc2)
16  (move_with_carrier robot carrier loc2 loc3)
17  (deliver_content robot box2 food carrier p3 loc3)
```

Figure 10: Plan generated by MBA + RGC*

Problem 2 and Problem 3 executions with MBA* and their generated plans follow.

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H hrgc -s mba domain.pddl problem2.pddl
2023-07-18 15:40:31,429 INFO     using search: mb_astar_search
2023-07-18 15:40:31,429 INFO     using heuristic: hRGCHeuristic
2023-07-18 15:40:31,430 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:40:31,431 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem2.pddl
2023-07-18 15:40:31,433 INFO     8 Predicates parsed
2023-07-18 15:40:31,433 INFO     5 Actions parsed
2023-07-18 15:40:31,433 INFO     25 Objects parsed
2023-07-18 15:40:31,433 INFO     0 Constants parsed
2023-07-18 15:40:31,433 INFO     Grounding start: emergency_aid
2023-07-18 15:40:31,453 INFO     Relevance analysis removed 0 facts
2023-07-18 15:40:31,454 INFO     Grounding end: emergency_aid
2023-07-18 15:40:31,454 INFO     132 Variables created
2023-07-18 15:40:31,454 INFO     1668 Operators created
2023-07-18 15:40:31,455 INFO     Search start: emergency_aid
2023-07-18 15:40:31,455 INFO     Initial h value: 12.000000
2023-07-18 15:40:31,685 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:40:31,899 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:40:32,130 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:40:44,275 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:40:44,864 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:40:44,927 INFO     Goal reached. Start extraction of solution.
2023-07-18 15:40:44,927 INFO     66022 Nodes expanded
2023-07-18 15:40:45,028 INFO     Search end: emergency_aid
2023-07-18 15:40:45,028 INFO     Search time: 1.4e+01
2023-07-18 15:40:45,028 INFO     Memory used: 0.1924 MB 195.2 MB
2023-07-18 15:40:45,030 INFO     Plan length: 74
```

Figure 11: Execution with MBA + RGC of problem 2*

problem2.pddl.soln

```
(fill_box robot1 box1 tools depot)
(load_box_on_carrier robot1 box1 tools carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l6)
(deliver_content robot1 box1 tools carrier2 p6 l6)
(fill_box robot2 box3 tools depot)
(load_box_on_carrier robot2 box3 tools carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l2)
(deliver_content robot2 box3 tools carrier1 p1 l2)
(move_with_carrier robot1 carrier2 l6 l5)
(move_with_carrier robot2 carrier1 l2 depot)
(fill_box robot2 box2 food depot)
(load_box_on_carrier robot2 box2 food carrier1 depot one two)
(move_with_carrier robot2 carrier1 depot l5)
(deliver_content robot1 box2 food carrier1 p5 l5)
(unload_box_from_carrier robot1 box3 carrier1 l5 two one)
(move_with_carrier robot1 carrier1 l5 l6)
(move_with_carrier robot2 carrier2 l5 l2)
(unload_box_from_carrier robot2 box1 carrier2 l2 one zero)
(move_with_carrier robot2 carrier2 l2 l6)
(move_with_carrier robot2 carrier1 l6 depot)
(unload_box_from_carrier robot2 box2 carrier1 depot one zero)
(fill_box robot2 box2 medicine depot)
(move_with_carrier robot1 carrier2 l6 depot)
(load_box_on_carrier robot1 box2 medicine carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l4)
(deliver_content robot1 box2 medicine carrier2 p4 l4)
(move_with_carrier robot1 carrier2 l4 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box2 medicine depot)
(load_box_on_carrier robot1 box2 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l6)
(deliver_content robot1 box2 medicine carrier1 p6 l6)
(move_with_carrier robot1 carrier1 l6 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 food depot)
(load_box_on_carrier robot1 box2 food carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l4)
(deliver_content robot1 box2 food carrier1 p4 l4)
(move_with_carrier robot1 carrier1 l4 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 tools depot)
(load_box_on_carrier robot1 box2 tools carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l5)
(deliver_content robot1 box2 tools carrier1 p5 l5)
(move_with_carrier robot1 carrier1 l5 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 food depot)
(load_box_on_carrier robot1 box2 food carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l6)
(deliver_content robot1 box2 food carrier1 p6 l6)
(move_with_carrier robot1 carrier1 l6 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 food depot)
(load_box_on_carrier robot1 box2 food carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l2)
(deliver_content robot1 box2 food carrier1 p1 l2)
(move_with_carrier robot1 carrier1 l2 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 medicine depot)
(load_box_on_carrier robot1 box2 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l5)
(deliver_content robot1 box2 medicine carrier1 p5 l5)
(move_with_carrier robot1 carrier1 l5 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 medicine depot)
(load_box_on_carrier robot1 box2 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l3)
(deliver_content robot1 box2 medicine carrier1 p3 l3)
(move_with_carrier robot1 carrier1 l3 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 medicine depot)
(load_box_on_carrier robot1 box2 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l2)
(deliver_content robot1 box2 medicine carrier1 p2 l2)
```

Figure 12: plan generated by MBA + RGC of problem 2*

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H hrgc -s mba domain.pddl problem3.pddl
2023-07-18 15:46:18,605 INFO     using search: mb_astar_search
2023-07-18 15:46:18,605 INFO     using heuristic: hRGCHeuristic
2023-07-18 15:46:18,605 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:46:18,606 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem3.pddl
2023-07-18 15:46:18,607 INFO     8 Predicates parsed
2023-07-18 15:46:18,607 INFO     5 Actions parsed
2023-07-18 15:46:18,607 INFO     30 Objects parsed
2023-07-18 15:46:18,607 INFO     0 Constants parsed
2023-07-18 15:46:18,607 INFO     Grounding start: emergency_aid
2023-07-18 15:46:18,642 INFO     Relevance analysis removed 0 facts
2023-07-18 15:46:18,643 INFO     Grounding end: emergency_aid
2023-07-18 15:46:18,643 INFO     200 Variables created
2023-07-18 15:46:18,643 INFO     3744 Operators created
2023-07-18 15:46:18,646 INFO     Search start: emergency_aid
2023-07-18 15:46:18,646 INFO     Initial h value: 18.000000
2023-07-18 15:46:19,055 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:46:19,463 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:46:19,888 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:46:58,057 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:46:58,767 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:46:59,925 INFO     Goal reached. Start extraction of solution.
2023-07-18 15:46:59,926 INFO     112724 Nodes expanded
2023-07-18 15:47:00,215 INFO     Search end: emergency_aid
2023-07-18 15:47:00,215 INFO     Search time: 4.2e+01
2023-07-18 15:47:00,215 INFO     Memory used: 0.1928 MB 615.5 MB
2023-07-18 15:47:00,218 INFO     Plan length: 117
```

Figure 13: Execution with MBA + RGC on problem 3*

problem3.pddl.soln

```
(fill_box robot2 box3 tools depot)
(load_box_on_carrier robot2 box3 tools carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l8)
(deliver_content robot2 box3 tools carrier1 p8 l8)
(fill_box robot1 box4 tools depot)
(load_box_on_carrier robot1 box4 tools carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l5)
(deliver_content robot1 box4 tools carrier2 p5 l5)
(move_with_carrier robot2 carrier1 l8 depot)
(unload_box_from_carrier robot2 box3 carrier1 depot one zero)
(move_with_carrier robot1 carrier2 l5 l4)
(fill_box robot2 box2 food depot)
(load_box_on_carrier robot2 box2 food carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l5)
(deliver_content robot2 box2 food carrier1 p5 l5)
(unload_box_from_carrier robot2 box2 carrier1 l5 one zero)
(move_with_carrier robot2 carrier1 l5 depot)
(fill_box robot2 box3 food depot)
(load_box_on_carrier robot2 box3 food carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l4)
(deliver_content robot2 box3 food carrier1 p4 l4)
(move_with_carrier robot2 carrier1 l4 depot)
(fill_box robot2 box1 food depot)
(move_with_carrier robot1 carrier2 l4 depot)
(unload_box_from_carrier robot2 box4 carrier2 depot one zero)
(fill_box robot2 box4 medicine depot)
(load_box_on_carrier robot2 box4 medicine carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l8)
(deliver_content robot1 box4 medicine carrier2 p8 l8)
(move_with_carrier robot2 carrier1 depot l5)
(move_with_carrier robot1 carrier2 l8 depot)
(load_box_on_carrier robot1 box1 food carrier2 depot one two)
(move_with_carrier robot1 carrier2 depot l2)
(deliver_content robot1 box1 food carrier2 p1 l2)
(move_with_carrier robot2 carrier1 l5 l7)
(unload_box_from_carrier robot2 box3 carrier1 l7 one zero)
(move_with_carrier robot2 carrier1 l7 l4)
(move_with_carrier robot1 carrier2 l2 depot)
(unload_box_from_carrier robot1 box1 carrier2 depot two one)
(fill_box robot1 box1 food depot)
(load_box_on_carrier robot1 box1 food carrier2 depot one two)
(move_with_carrier robot1 carrier2 depot l8)
(deliver_content robot1 box1 food carrier2 p8 l8)
(move_with_carrier robot2 carrier1 l4 depot)
(move_with_carrier robot1 carrier2 l8 depot)
(unload_box_from_carrier robot2 box4 carrier2 depot two one)
(fill_box robot2 box4 food depot)
(load_box_on_carrier robot2 box4 food carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l6)
(deliver_content robot2 box4 food carrier1 p6 l6)
(unload_box_from_carrier robot1 box1 carrier2 depot one zero)
(fill_box robot1 box1 food depot)
(load_box_on_carrier robot1 box1 food carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l7)
(deliver_content robot1 box1 food carrier2 p7 l7)
(unload_box_from_carrier robot1 box1 carrier2 l7 one zero)
(move_with_carrier robot2 carrier1 l6 depot)
(unload_box_from_carrier robot2 box4 carrier1 depot one zero)
(fill_box robot2 box4 medicine depot)
(move_with_carrier robot1 carrier2 l7 depot)
(load_box_on_carrier robot2 box4 medicine carrier2 depot zero one)
(move_with_carrier robot2 carrier2 depot l5)
(deliver_content robot2 box4 medicine carrier2 p5 l5)
(move_with_carrier robot2 carrier2 l5 depot)
(move_with_carrier robot1 carrier1 depot l2)
(unload_box_from_carrier robot2 box4 carrier2 depot one zero)
(fill_box robot2 box4 tools depot)
(load_box_on_carrier robot2 box4 tools carrier2 depot zero one)
(move_with_carrier robot2 carrier2 depot l6)
(deliver_content robot2 box4 tools carrier2 p6 l6)
(move_with_carrier robot2 carrier2 l6 depot)
(unload_box_from_carrier robot2 box4 carrier2 depot one zero)
(fill_box robot2 box4 tools depot)
(move_with_carrier robot1 carrier1 l2 depot)
(load_box_on_carrier robot2 box4 tools carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l7)
(deliver_content robot1 box4 tools carrier1 p7 l7)
(move_with_carrier robot2 carrier2 depot l2)
(move_with_carrier robot1 carrier1 l7 depot)
(unload_box_from_carrier robot1 box4 carrier1 depot one zero)
(fill_box robot1 box4 medicine depot)
(load_box_on_carrier robot1 box4 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l4)
(deliver_content robot1 box4 medicine carrier1 p4 l4)
(move_with_carrier robot1 carrier1 l4 depot)
(unload_box_from_carrier robot1 box4 carrier1 depot one zero)
(fill_box robot1 box4 medicine depot)
(move_with_carrier robot2 carrier2 l2 depot)
(load_box_on_carrier robot2 box4 medicine carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l6)
(deliver_content robot1 box4 medicine carrier2 p6 l6)
(move_with_carrier robot2 carrier1 depot l2)
(move_with_carrier robot1 carrier2 l6 depot)
(unload_box_from_carrier robot1 box4 carrier2 depot one zero)
(fill_box robot1 box4 medicine depot)
(load_box_on_carrier robot1 box4 medicine carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l3)
(deliver_content robot1 box4 medicine carrier2 p3 l3)
(move_with_carrier robot1 carrier2 l3 depot)
(unload_box_from_carrier robot1 box4 carrier2 depot one zero)
(fill_box robot1 box4 medicine depot)
(move_with_carrier robot2 carrier1 l2 depot)
(load_box_on_carrier robot2 box4 medicine carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l7)
(deliver_content robot2 box4 medicine carrier1 p7 l7)
(move_with_carrier robot2 carrier1 l7 depot)
(unload_box_from_carrier robot2 box4 carrier1 depot one zero)
(fill_box robot2 box4 tools depot)
(load_box_on_carrier robot2 box4 tools carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l2)
(deliver_content robot2 box4 tools carrier1 p1 l2)
(move_with_carrier robot2 carrier1 l2 depot)
(unload_box_from_carrier robot2 box4 carrier1 depot one zero)
(fill_box robot2 box4 medicine depot)
(load_box_on_carrier robot2 box4 medicine carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l2)
(deliver_content robot2 box4 medicine carrier1 p2 l2)
```

Figure 14: Plan generated by MBA + RGC on problem 3*

HAMMING

This heuristic estimates the number of differences between the current state and the target state by calculating the Hamming distance.

```python
class hHammingHeuristic(_RelaxationHeuristic):

    """
    L'euristica Hamming stima il numero di differenze tra lo stato corrente e lo stato obiettivo calcolando la distanza di Hamming:
    si tratta del conteggio delle posizioni in cui gli elementi corrispondenti in due stati sono diversi.
    """

    def __init__(self, task): #Inizializzazione
        super().__init__(task)

    def __call__(self, node):
        current_state = set(node.state)
        goal_state = set(self.goals)

        h_value = sum(current_fact != goal_fact for current_fact, goal_fact in zip(current_state, goal_state)) #Il valore restituito è la somma
                                                                                                               #dei confronti tra gli elementi di
                                                                                                               #current_state e goal_stat

        return h_value
```

Figure 15: class hHammingHeuristic (present in the relaxation.py file)

In the __call__ method, the value of the heuristic is calculated in this way:

- The current state of the node is obtained and a set called current_state is created containing the elements of the current state.
- A set called goal_state is created which contains the elements of the goal state.
- The h_value is calculated as the sum of the comparisons of the elements of current_state and goal_state. The zip function is used to iterate simultaneously over the elements of the two lists.
- Within the loop, each current element is compared with the corresponding goal element. If the two elements are different (i.e. the comparison current_fact != goal_fact is true), h_value is incremented.
- At the end of the loop, h_value will contain the number of elements that differ between current_state and goal_state.

Again, the plan was generated with both A* and MBA*. The results are in line with those previously obtained. As expected, MBA* reduces memory consumption and improves search times, at the expense of plan length.

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H hhamming -s mba domain.pddl problem1.pddl
2023-07-18 15:49:55,445 INFO     using search: mb_astar_search
2023-07-18 15:49:55,445 INFO     using heuristic: hHammingHeuristic
2023-07-18 15:49:55,445 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:49:55,446 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem1.pddl
2023-07-18 15:49:55,446 INFO     8 Predicates parsed
2023-07-18 15:49:55,446 INFO     5 Actions parsed
2023-07-18 15:49:55,446 INFO     20 Objects parsed
2023-07-18 15:49:55,446 INFO     0 Constants parsed
2023-07-18 15:49:55,446 INFO     Grounding start: emergency_aid
2023-07-18 15:49:55,451 INFO     Relevance analysis removed 0 facts
2023-07-18 15:49:55,451 INFO     Grounding end: emergency_aid
2023-07-18 15:49:55,451 INFO     65 Variables created
2023-07-18 15:49:55,451 INFO     276 Operators created
2023-07-18 15:49:55,451 INFO     Search start: emergency_aid
2023-07-18 15:49:55,451 INFO     Initial h value: 4.000000
2023-07-18 15:49:55,523 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:49:55,585 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:49:55,648 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:49:56,619 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:49:56,691 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:49:56,710 INFO     Goal reached. Start extraction of solution.
2023-07-18 15:49:56,710 INFO     17246 Nodes expanded
2023-07-18 15:49:56,732 INFO     Search end: emergency_aid
2023-07-18 15:49:56,732 INFO     Search time: 1.3
2023-07-18 15:49:56,732 INFO     Memory used: 0.1917 MB 50.54 MB
2023-07-18 15:49:56,732 INFO     Plan length: 19
```

Figure 16: Execution with MBA + Hamming on problem 1*

problem1.pddl.soln

```
(fill_box robot box4 food depot)
(fill_box robot box3 food depot)
(load_box_on_carrier robot box4 food carrier depot zero one)
(move_with_carrier robot carrier depot loc3)
(deliver_content robot box4 food carrier p3 loc3)
(unload_box_from_carrier robot box4 carrier loc3 one zero)
(move_with_carrier robot carrier loc3 depot)
(fill_box robot box2 food depot)
(fill_box robot box1 medicine depot)
(fill_box robot box5 medicine depot)
(load_box_on_carrier robot box1 medicine carrier depot zero one)
(load_box_on_carrier robot box5 medicine carrier depot one two)
(move_with_carrier robot carrier depot loc2)
(deliver_content robot box1 medicine carrier p2 loc2)
(deliver_content robot box5 medicine carrier p1 loc2)
(move_with_carrier robot carrier loc2 depot)
(load_box_on_carrier robot box2 food carrier depot two three)
(move_with_carrier robot carrier depot loc2)
(deliver_content robot box2 food carrier p1 loc2)
```

Figure 17: MBA + Hamming generated plan on problem 1*

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H hhamming -s astar domain.pddl problem1.pddl
2023-07-18 16:26:14,754 INFO     using search: astar_search
2023-07-18 16:26:14,754 INFO     using heuristic: hHammingHeuristic
2023-07-18 16:26:14,754 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 16:26:14,755 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem1.pddl
2023-07-18 16:26:14,756 INFO     8 Predicates parsed
2023-07-18 16:26:14,756 INFO     5 Actions parsed
2023-07-18 16:26:14,756 INFO     20 Objects parsed
2023-07-18 16:26:14,756 INFO     0 Constants parsed
2023-07-18 16:26:14,756 INFO     Grounding start: emergency_aid
2023-07-18 16:26:14,760 INFO     Relevance analysis removed 0 facts
2023-07-18 16:26:14,761 INFO     Grounding end: emergency_aid
2023-07-18 16:26:14,761 INFO     65 Variables created
2023-07-18 16:26:14,761 INFO     276 Operators created
2023-07-18 16:26:14,761 INFO     Search start: emergency_aid
2023-07-18 16:26:14,761 INFO     Initial h value: 4.000000
2023-07-18 16:26:37,023 INFO     Goal reached. Start extraction of solution.
2023-07-18 16:26:37,023 INFO     339134 Nodes expanded
2023-07-18 16:26:37,751 INFO     Search end: emergency_aid
2023-07-18 16:26:37,751 INFO     Search time: 2.3e+01
2023-07-18 16:26:37,751 INFO     Memory used: 0.1389 MB 641.3 MB
2023-07-18 16:26:37,753 INFO     Plan length: 14
```

Figure 18: Execution with A + Hamming of problem 1*

problem1.pddl.soln

```
(fill_box robot box4 medicine depot)
(fill_box robot box3 medicine depot)
(fill_box robot box1 food depot)
(fill_box robot box2 food depot)
(load_box_on_carrier robot box4 medicine carrier depot zero one)
(load_box_on_carrier robot box3 medicine carrier depot one two)
(load_box_on_carrier robot box1 food carrier depot two three)
(load_box_on_carrier robot box2 food carrier depot three four)
(move_with_carrier robot carrier depot loc2)
(deliver_content robot box1 food carrier p1 loc2)
(deliver_content robot box4 medicine carrier p1 loc2)
(deliver_content robot box3 medicine carrier p2 loc2)
(move_with_carrier robot carrier loc2 loc3)
(deliver_content robot box2 food carrier p3 loc3)
```

Figure 19: plane generated by A + Hamming on problem 1*

The plan on problem 2 was also generated for this heuristic. Comparing the results obtained with the plan generated on the same problem and with the Relaxed Goal Count heuristic, one can see differences in the memory used and execution time:

	MEMORY	SEARCH TIME	PLAN LENGTH
HAMMING	293.1 MB	19 s	79
RGC	195.2 MB	14 s	74

The Hamming heuristic calculates the distance between the current state and the target state by counting the number of different positions between the two states, which can be very computationally burdensome.

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H hhamming -s mba domain.pddl problem2.pddl
2023-07-18 15:51:45,498 INFO     using search: mb_astar_search
2023-07-18 15:51:45,498 INFO     using heuristic: hHammingHeuristic
2023-07-18 15:51:45,498 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:51:45,499 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem2.pddl
2023-07-18 15:51:45,499 INFO     8 Predicates parsed
2023-07-18 15:51:45,499 INFO     5 Actions parsed
2023-07-18 15:51:45,499 INFO     25 Objects parsed
2023-07-18 15:51:45,499 INFO     0 Constants parsed
2023-07-18 15:51:45,499 INFO     Grounding start: emergency_aid
2023-07-18 15:51:45,517 INFO     Relevance analysis removed 0 facts
2023-07-18 15:51:45,517 INFO     Grounding end: emergency_aid
2023-07-18 15:51:45,517 INFO     132 Variables created
2023-07-18 15:51:45,517 INFO     1668 Operators created
2023-07-18 15:51:45,518 INFO     Search start: emergency_aid
2023-07-18 15:51:45,518 INFO     Initial h value: 12.000000
2023-07-18 15:51:45,760 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:51:45,984 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:51:46,206 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:52:04,319 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:52:04,513 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:52:04,690 INFO     Goal reached. Start extraction of solution.
2023-07-18 15:52:04,690 INFO     88241 Nodes expanded
2023-07-18 15:52:04,864 INFO     Search end: emergency_aid
2023-07-18 15:52:04,864 INFO     Search time: 1.9e+01
2023-07-18 15:52:04,864 INFO     Memory used: 0.1926 MB 293.1 MB
2023-07-18 15:52:04,866 INFO     Plan length: 79
```

Figure 20: MBA + Hamming execution of problem 2

problem2.pddl.soln

```
(fill_box robot1 box3 tools depot)
(fill_box robot1 box2 medicine depot)
(load_box_on_carrier robot1 box3 tools carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l3)
(fill_box robot2 box1 medicine depot)
(load_box_on_carrier robot2 box1 medicine carrier2 depot zero one)
(move_with_carrier robot1 carrier1 l3 depot)
(move_with_carrier robot2 carrier1 depot l5)
(load_box_on_carrier robot1 box2 medicine carrier2 depot one two)
(deliver_content robot2 box3 tools carrier1 p5 l5)
(move_with_carrier robot2 carrier1 l5 l4)
(move_with_carrier robot1 carrier2 depot l2)
(deliver_content robot1 box1 medicine carrier2 p2 l2)
(unload_box_from_carrier robot1 box1 carrier2 l2 two one)
(move_with_carrier robot1 carrier2 l2 l3)
(move_with_carrier robot2 carrier1 l4 l5)
(deliver_content robot1 box2 medicine carrier2 p3 l3)
(unload_box_from_carrier robot1 box2 carrier2 l3 one zero)
(move_with_carrier robot1 carrier2 l3 l5)
(move_with_carrier robot1 carrier1 l5 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(fill_box robot1 box3 tools depot)
(load_box_on_carrier robot1 box3 tools carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l2)
(move_with_carrier robot2 carrier2 l5 depot)
(deliver_content robot1 box3 tools carrier1 p1 l2)
(move_with_carrier robot1 carrier1 l2 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(fill_box robot1 box3 medicine depot)
(load_box_on_carrier robot1 box3 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l6)
(move_with_carrier robot2 carrier2 depot l6)
(deliver_content robot1 box3 medicine carrier1 p6 l6)
(move_with_carrier robot2 carrier1 l6 depot)
(unload_box_from_carrier robot2 box3 carrier1 depot one zero)
(fill_box robot2 box3 food depot)
(load_box_on_carrier robot2 box3 food carrier1 depot zero one)
(move_with_carrier robot1 carrier2 l6 l4)
(move_with_carrier robot2 carrier1 depot l4)
(deliver_content robot1 box3 food carrier1 p4 l4)
(move_with_carrier robot1 carrier1 l4 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(fill_box robot1 box3 medicine depot)
(load_box_on_carrier robot1 box3 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l4)
(move_with_carrier robot1 carrier2 l4 depot)
(deliver_content robot2 box3 medicine carrier1 p4 l4)
(move_with_carrier robot2 carrier1 l4 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(fill_box robot1 box3 medicine depot)
(load_box_on_carrier robot1 box3 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l5)
(move_with_carrier robot2 carrier2 depot l2)
(deliver_content robot1 box3 medicine carrier1 p5 l5)
(move_with_carrier robot1 carrier1 l5 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(fill_box robot1 box3 food depot)
(load_box_on_carrier robot1 box3 food carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l5)
(deliver_content robot1 box3 food carrier1 p5 l5)
(move_with_carrier robot1 carrier1 l5 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(fill_box robot1 box3 tools depot)
(load_box_on_carrier robot1 box3 tools carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l6)
(deliver_content robot1 box3 tools carrier1 p6 l6)
(move_with_carrier robot1 carrier1 l6 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(fill_box robot1 box3 food depot)
(load_box_on_carrier robot1 box3 food carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l6)
(move_with_carrier robot2 carrier2 l2 depot)
(deliver_content robot1 box3 food carrier1 p6 l6)
(move_with_carrier robot1 carrier1 l6 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(fill_box robot1 box3 food depot)
(load_box_on_carrier robot1 box3 food carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l2)
(deliver_content robot1 box3 food carrier1 p1 l2)
```

Figure 21: Plan generated by MBA + Hamming of problem 2*

This is followed by a comparison of the RGC heuristic and the Hamming heuristic on problem 3.

	MEMORY	SEARCH TIME	PLAN LENGTH
HAMMING	797.7 MB	54 s	119
RGC	615 MB	42 s	117

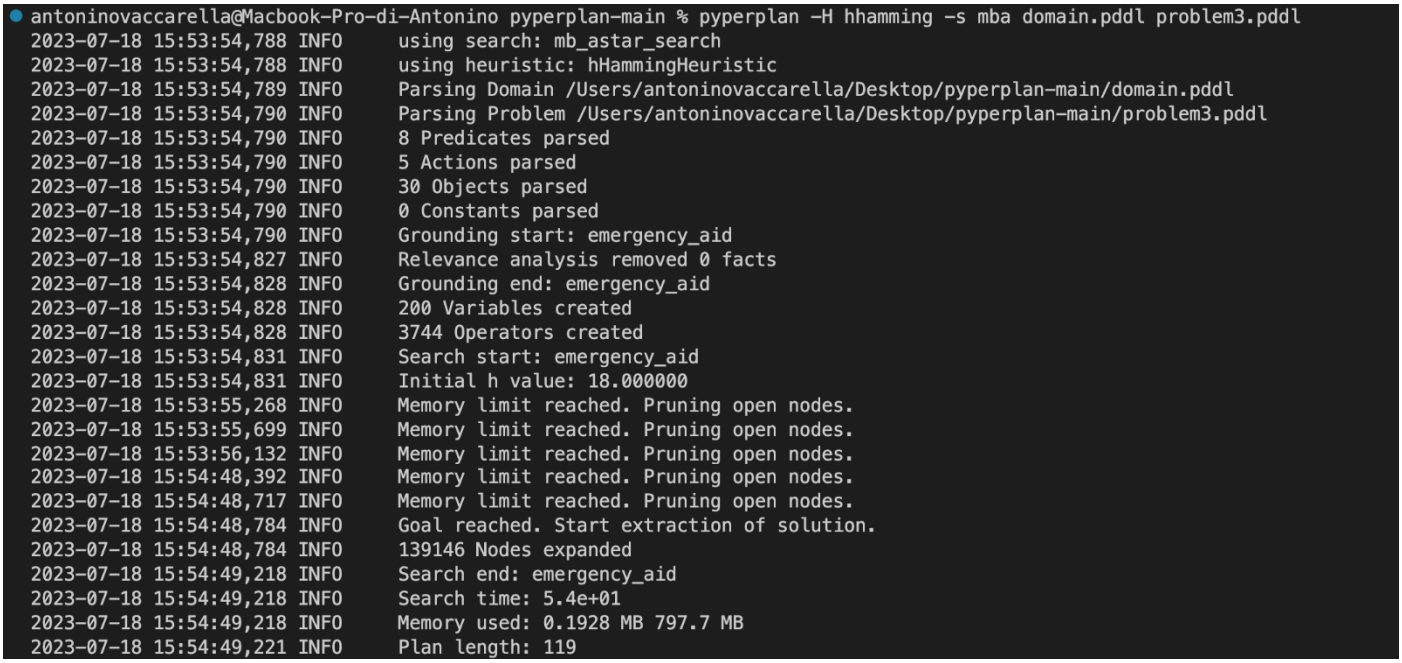

Figure 22: MBA + Hamming execution of problem 3*

problem3.pddl.soln

```
(fill_box robot2 box4 food depot)
(fill_box robot2 box1 medicine depot)
(fill_box robot2 box3 food depot)
(load_box_on_carrier robot2 box4 food carrier1 depot zero one)
(load_box_on_carrier robot2 box3 food carrier2 depot zero one)
(move_with_carrier robot2 carrier1 depot l4)
(deliver_content robot2 box4 food carrier1 p4 l4)
(move_with_carrier robot2 carrier1 l4 l7)
(unload_box_from_carrier robot2 box4 carrier1 l7 one zero)
(move_with_carrier robot2 carrier1 l7 depot)
(move_with_carrier robot1 carrier2 depot l2)
(fill_box robot2 box2 medicine depot)
(deliver_content robot1 box3 food carrier2 p1 l2)
(move_with_carrier robot1 carrier2 l2 depot)
(load_box_on_carrier robot2 box1 medicine carrier2 depot one two)
(move_with_carrier robot2 carrier2 depot l2)
(deliver_content robot2 box1 medicine carrier2 p2 l2)
(unload_box_from_carrier robot2 box3 carrier2 l2 two one)
(move_with_carrier robot2 carrier2 l2 depot)
(load_box_on_carrier robot2 box2 medicine carrier2 depot one two)
(move_with_carrier robot2 carrier1 depot l7)
(move_with_carrier robot1 carrier2 depot l5)
(unload_box_from_carrier robot1 box1 carrier2 l5 two one)
(deliver_content robot1 box2 medicine carrier2 p5 l5)
(move_with_carrier robot1 carrier2 l5 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box2 medicine depot)
(move_with_carrier robot2 carrier1 l7 depot)
(load_box_on_carrier robot2 box2 medicine carrier1 depot zero one)
(move_with_carrier robot2 carrier2 depot l2)
(move_with_carrier robot1 carrier1 depot l4)
(deliver_content robot1 box2 medicine carrier1 p4 l4)
(move_with_carrier robot1 carrier1 l4 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 medicine depot)
(load_box_on_carrier robot1 box2 medicine carrier1 depot zero one)
(move_with_carrier robot2 carrier2 l2 depot)
(move_with_carrier robot2 carrier1 depot l7)
(deliver_content robot2 box2 medicine carrier1 p7 l7)
(move_with_carrier robot2 carrier1 l7 depot)
(unload_box_from_carrier robot2 box2 carrier1 depot one zero)
(fill_box robot2 box2 food depot)
(load_box_on_carrier robot2 box2 food carrier2 depot zero one)
(move_with_carrier robot2 carrier1 depot l2)
(move_with_carrier robot1 carrier2 depot l6)
(deliver_content robot1 box2 food carrier2 p6 l6)
(move_with_carrier robot1 carrier2 l6 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box2 food depot)
(load_box_on_carrier robot1 box2 food carrier2 depot zero one)
(move_with_carrier robot2 carrier1 l2 l5)
(move_with_carrier robot1 carrier2 depot l5)
(deliver_content robot2 box2 food carrier2 p5 l5)
(move_with_carrier robot2 carrier2 l5 depot)
(unload_box_from_carrier robot2 box2 carrier2 depot one zero)
(fill_box robot2 box2 tools depot)
(move_with_carrier robot2 carrier2 depot l5)
(move_with_carrier robot1 carrier1 l5 depot)
(load_box_on_carrier robot1 box2 tools carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l6)
(deliver_content robot1 box2 tools carrier1 p6 l6)
(move_with_carrier robot1 carrier1 l6 depot)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(fill_box robot1 box2 tools depot)
(load_box_on_carrier robot1 box2 tools carrier1 depot zero one)
(move_with_carrier robot2 carrier2 l5 depot)
(move_with_carrier robot2 carrier1 depot l8)
(deliver_content robot2 box2 tools carrier1 p8 l8)
(move_with_carrier robot2 carrier1 l8 depot)
(unload_box_from_carrier robot2 box2 carrier1 depot one zero)
(fill_box robot2 box2 food depot)
(load_box_on_carrier robot2 box2 food carrier2 depot zero one)
(move_with_carrier robot2 carrier2 depot l8)
(move_with_carrier robot1 carrier1 depot l8)
(deliver_content robot2 box2 food carrier2 p8 l8)
(move_with_carrier robot2 carrier2 l8 depot)
(unload_box_from_carrier robot2 box2 carrier2 depot one zero)
(fill_box robot2 box2 food depot)
(load_box_on_carrier robot2 box2 food carrier2 depot zero one)
(move_with_carrier robot2 carrier2 depot l7)
(deliver_content robot2 box2 food carrier2 p7 l7)
(move_with_carrier robot2 carrier2 l7 depot)
(unload_box_from_carrier robot2 box2 carrier2 depot one zero)
(fill_box robot2 box2 medicine depot)
(load_box_on_carrier robot2 box2 medicine carrier2 depot zero one)
(move_with_carrier robot2 carrier2 depot l3)
(move_with_carrier robot1 carrier1 l8 l3)
(deliver_content robot2 box2 medicine carrier2 p3 l3)
(move_with_carrier robot1 carrier2 l3 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box2 medicine depot)
(load_box_on_carrier robot1 box2 medicine carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l8)
(deliver_content robot1 box2 medicine carrier2 p8 l8)
(move_with_carrier robot1 carrier2 l8 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box2 tools depot)
(load_box_on_carrier robot1 box2 tools carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l2)
(move_with_carrier robot2 carrier1 l3 l8)
(deliver_content robot1 box2 tools carrier2 p1 l2)
(move_with_carrier robot1 carrier2 l2 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box2 tools depot)
(load_box_on_carrier robot1 box2 tools carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l5)
(deliver_content robot1 box2 tools carrier2 p5 l5)
(move_with_carrier robot1 carrier2 l5 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box2 medicine depot)
(load_box_on_carrier robot1 box2 medicine carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l6)
(deliver_content robot1 box2 medicine carrier2 p6 l6)
(move_with_carrier robot1 carrier2 l6 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box2 tools depot)
(load_box_on_carrier robot1 box2 tools carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l7)
(deliver_content robot1 box2 tools carrier2 p7 l7)
```

Figure 23: Plan generated by MBA + Hamming on problem 3*

RANDOM

This class implements the Random heuristic. It was implemented because it provided a basis of comparison for evaluating the efficiency of the other heuristics.

```python
class RandomHeuristic(_RelaxationHeuristic):

    def __init__(self, task): #Inizializzazione
        super().__init__(task)

    def __call__(self, node):
        return random.randint(0, 100) #Viene restituito un valore random compreso tra (0 e 100) per ogni stato
```

Figure 24: RandomHeuristic class (found in the relaxation.py file)

The heuristic does not take into account the characteristics of the current state or the target state, but simply returns a random value between 0 and 100. It was useful for testing previous heuristics and verifying their behaviour against the random heuristic on the same problem instances.

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H random -s mba domain.pddl problem1.pddl
2023-07-18 15:57:27,457 INFO     using search: mb_astar_search
2023-07-18 15:57:27,457 INFO     using heuristic: RandomHeuristic
2023-07-18 15:57:27,457 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:57:27,458 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem1.pddl
2023-07-18 15:57:27,459 INFO     8 Predicates parsed
2023-07-18 15:57:27,459 INFO     5 Actions parsed
2023-07-18 15:57:27,459 INFO     20 Objects parsed
2023-07-18 15:57:27,459 INFO     0 Constants parsed
2023-07-18 15:57:27,459 INFO     Grounding start: emergency_aid
2023-07-18 15:57:27,463 INFO     Relevance analysis removed 0 facts
2023-07-18 15:57:27,463 INFO     Grounding end: emergency_aid
2023-07-18 15:57:27,463 INFO     65 Variables created
2023-07-18 15:57:27,463 INFO     276 Operators created
2023-07-18 15:57:27,464 INFO     Search start: emergency_aid
2023-07-18 15:57:27,464 INFO     Initial h value: 88.000000
2023-07-18 15:57:27,522 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:57:27,574 INFO     Goal reached. Start extraction of solution.
2023-07-18 15:57:27,574 INFO     1841 Nodes expanded
2023-07-18 15:57:27,579 INFO     Search end: emergency_aid
2023-07-18 15:57:27,579 INFO     Search time: 0.12
2023-07-18 15:57:27,579 INFO     Memory used: 0.1895 MB 9.802 MB
2023-07-18 15:57:27,580 INFO     Plan length: 22
```

Figure 25: Execution with MBA + Random of problem 1 (h_value = 88)*

problem1.pddl.soln

```
(fill_box robot box5 medicine depot)
(fill_box robot box1 food depot)
(load_box_on_carrier robot box1 food carrier depot zero one)
(fill_box robot box3 medicine depot)
(fill_box robot box2 food depot)
(load_box_on_carrier robot box5 medicine carrier depot one two)
(load_box_on_carrier robot box3 medicine carrier depot two three)
(fill_box robot box4 medicine depot)
(move_with_carrier robot carrier depot loc3)
(deliver_content robot box1 food carrier p3 loc3)
(move_with_carrier robot carrier loc3 loc2)
(deliver_content robot box5 medicine carrier p1 loc2)
(move_with_carrier robot carrier loc2 depot)
(load_box_on_carrier robot box2 food carrier depot three four)
(unload_box_from_carrier robot box5 carrier depot four three)
(unload_box_from_carrier robot box1 carrier depot three two)
(fill_box robot box1 medicine depot)
(fill_box robot box5 medicine depot)
(load_box_on_carrier robot box5 medicine carrier depot two three)
(move_with_carrier robot carrier depot loc2)
(deliver_content robot box3 medicine carrier p2 loc2)
(deliver_content robot box2 food carrier p1 loc2)
```

*Figure 26: Plan generated by Memory Bounded A**

As can be seen from the images below, the behaviour of the algorithm with this heuristic is totally random and varies regardless of the initial value of the heuristic.

```
antoninovaccarella@MBP-di-Antonino pyperplan-main % pyperplan -H random -s mba domain.pddl problem1.pddl
2023-07-19 09:58:34,347 INFO     using search: mb_astar_search
2023-07-19 09:58:34,347 INFO     using heuristic: RandomHeuristic
2023-07-19 09:58:34,347 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-19 09:58:34,348 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem1.pddl
2023-07-19 09:58:34,349 INFO     8 Predicates parsed
2023-07-19 09:58:34,349 INFO     5 Actions parsed
2023-07-19 09:58:34,349 INFO     20 Objects parsed
2023-07-19 09:58:34,349 INFO     0 Constants parsed
2023-07-19 09:58:34,349 INFO     Grounding start: emergency_aid
2023-07-19 09:58:34,353 INFO     Relevance analysis removed 0 facts
2023-07-19 09:58:34,353 INFO     Grounding end: emergency_aid
2023-07-19 09:58:34,353 INFO     65 Variables created
2023-07-19 09:58:34,353 INFO     276 Operators created
2023-07-19 09:58:34,354 INFO     Search start: emergency_aid
2023-07-19 09:58:34,354 INFO     Initial h value: 13.000000
2023-07-19 09:58:34,451 INFO     Goal reached. Start extraction of solution.
2023-07-19 09:58:34,451 INFO     1716 Nodes expanded
2023-07-19 09:58:34,456 INFO     Search end: emergency_aid
2023-07-19 09:58:34,456 INFO     Search time: 0.1
2023-07-19 09:58:34,456 INFO     Memory used: 0.1379 MB 9.635 MB
2023-07-19 09:58:34,457 INFO     Plan length: 20
```

Figure 27: Execution with MBA + Random of problem 1 (h_value = 13)*

```
antoninovaccarella@MBP-di-Antonino pyperplan-main % pyperplan -H random -s mba domain.pddl problem1.pddl
2023-07-19 09:59:26,713 INFO     using search: mb_astar_search
2023-07-19 09:59:26,713 INFO     using heuristic: RandomHeuristic
2023-07-19 09:59:26,713 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-19 09:59:26,714 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem1.pddl
2023-07-19 09:59:26,715 INFO     8 Predicates parsed
2023-07-19 09:59:26,715 INFO     5 Actions parsed
2023-07-19 09:59:26,715 INFO     20 Objects parsed
2023-07-19 09:59:26,715 INFO     0 Constants parsed
2023-07-19 09:59:26,715 INFO     Grounding start: emergency_aid
2023-07-19 09:59:26,719 INFO     Relevance analysis removed 0 facts
2023-07-19 09:59:26,719 INFO     Grounding end: emergency_aid
2023-07-19 09:59:26,719 INFO     65 Variables created
2023-07-19 09:59:26,719 INFO     276 Operators created
2023-07-19 09:59:26,720 INFO     Search start: emergency_aid
2023-07-19 09:59:26,720 INFO     Initial h value: 26.000000
2023-07-19 09:59:26,808 INFO     Goal reached. Start extraction of solution.
2023-07-19 09:59:26,808 INFO     1531 Nodes expanded
2023-07-19 09:59:26,813 INFO     Search end: emergency_aid
2023-07-19 09:59:26,813 INFO     Search time: 0.093
2023-07-19 09:59:26,813 INFO     Memory used: 0.1378 MB 9.002 MB
2023-07-19 09:59:26,814 INFO     Plan length: 17
```

Figure 28: Execution with MBA + Random of problem 1 (h_value = 26)*

```
antoninovaccarella@MBP-di-Antonino pyperplan-main % pyperplan -H random -s mba domain.pddl problem1.pddl
2023-07-19 09:53:05,082 INFO     using search: mb_astar_search
2023-07-19 09:53:05,082 INFO     using heuristic: RandomHeuristic
2023-07-19 09:53:05,082 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-19 09:53:05,085 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem1.pddl
2023-07-19 09:53:05,087 INFO     8 Predicates parsed
2023-07-19 09:53:05,087 INFO     5 Actions parsed
2023-07-19 09:53:05,087 INFO     20 Objects parsed
2023-07-19 09:53:05,087 INFO     0 Constants parsed
2023-07-19 09:53:05,087 INFO     Grounding start: emergency_aid
2023-07-19 09:53:05,092 INFO     Relevance analysis removed 0 facts
2023-07-19 09:53:05,092 INFO     Grounding end: emergency_aid
2023-07-19 09:53:05,092 INFO     65 Variables created
2023-07-19 09:53:05,092 INFO     276 Operators created
2023-07-19 09:53:05,093 INFO     Search start: emergency_aid
2023-07-19 09:53:05,093 INFO     Initial h value: 67.000000
2023-07-19 09:53:05,200 INFO     Goal reached. Start extraction of solution.
2023-07-19 09:53:05,200 INFO     1938 Nodes expanded
2023-07-19 09:53:05,206 INFO     Search end: emergency_aid
2023-07-19 09:53:05,206 INFO     Search time: 0.11
2023-07-19 09:53:05,206 INFO     Memory used: 0.1378 MB 10.74 MB
2023-07-19 09:53:05,207 INFO     Plan length: 22
```

Figure 29: Execution with MBA + Random of problem 1 (h_value = 67)*

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H random -s mba domain.pddl problem2.pddl
2023-07-18 15:59:22,503 INFO     using search: mb_astar_search
2023-07-18 15:59:22,503 INFO     using heuristic: RandomHeuristic
2023-07-18 15:59:22,503 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:59:22,505 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem2.pddl
2023-07-18 15:59:22,506 INFO     8 Predicates parsed
2023-07-18 15:59:22,506 INFO     5 Actions parsed
2023-07-18 15:59:22,506 INFO     25 Objects parsed
2023-07-18 15:59:22,506 INFO     0 Constants parsed
2023-07-18 15:59:22,506 INFO     Grounding start: emergency_aid
2023-07-18 15:59:22,524 INFO     Relevance analysis removed 0 facts
2023-07-18 15:59:22,524 INFO     Grounding end: emergency_aid
2023-07-18 15:59:22,524 INFO     132 Variables created
2023-07-18 15:59:22,524 INFO     1668 Operators created
2023-07-18 15:59:22,525 INFO     Search start: emergency_aid
2023-07-18 15:59:22,526 INFO     Initial h value: 53.000000
2023-07-18 15:59:22,708 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:59:22,899 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:59:23,089 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 16:00:25,975 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 16:00:26,183 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 16:00:26,363 INFO     Goal reached. Start extraction of solution.
2023-07-18 16:00:26,363 INFO     265129 Nodes expanded
2023-07-18 16:00:27,701 INFO     Search end: emergency_aid
2023-07-18 16:00:27,701 INFO     Search time: 6.5e+01
2023-07-18 16:00:27,701 INFO     Memory used: 0.193 MB 2.104e+03 MB
2023-07-18 16:00:27,703 INFO     Plan length: 131
```

Figure 30: Execution with MBA + Random of problem 2*

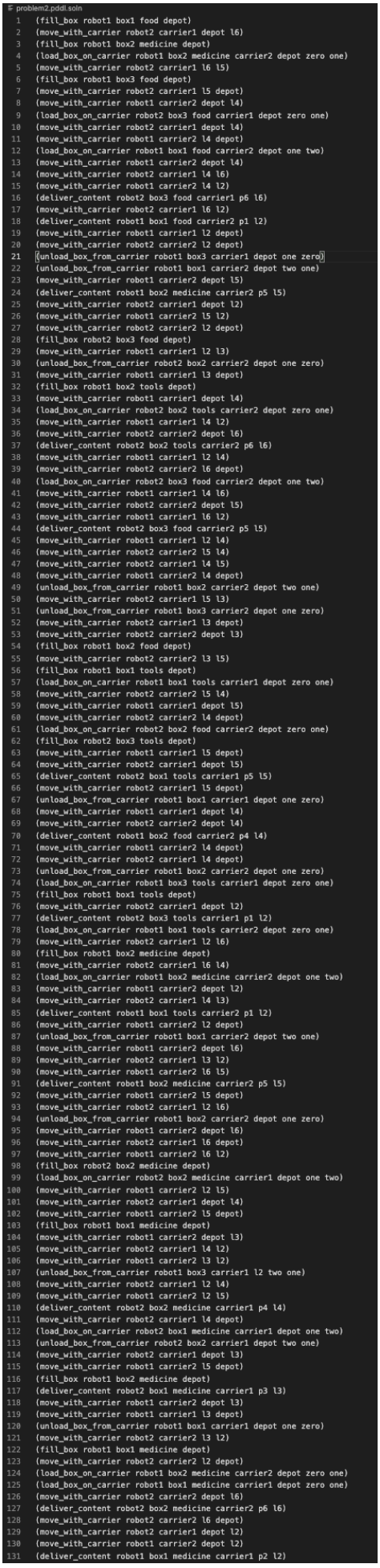

Figure 31: Plan generated by MBA + Random with problem 2*

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H random -s mba domain.pddl problem3.pddl
2023-07-18 16:06:35,008 INFO     using search: mb_astar_search
2023-07-18 16:06:35,008 INFO     using heuristic: RandomHeuristic
2023-07-18 16:06:35,008 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 16:06:35,009 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem3.pddl
2023-07-18 16:06:35,010 INFO     8 Predicates parsed
2023-07-18 16:06:35,010 INFO     5 Actions parsed
2023-07-18 16:06:35,010 INFO     30 Objects parsed
2023-07-18 16:06:35,010 INFO     0 Constants parsed
2023-07-18 16:06:35,010 INFO     Grounding start: emergency_aid
2023-07-18 16:06:35,045 INFO     Relevance analysis removed 0 facts
2023-07-18 16:06:35,046 INFO     Grounding end: emergency_aid
2023-07-18 16:06:35,046 INFO     200 Variables created
2023-07-18 16:06:35,046 INFO     3744 Operators created
2023-07-18 16:06:35,048 INFO     Search start: emergency_aid
2023-07-18 16:06:35,048 INFO     Initial h value: 84.000000
2023-07-18 16:06:35,382 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 16:06:35,743 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 16:06:36,111 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 16:10:33,519 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 16:10:33,879 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 16:10:33,927 INFO     Goal reached. Start extraction of solution.
2023-07-18 16:10:33,927 INFO     557906 Nodes expanded
2023-07-18 16:10:39,112 INFO     Search end: emergency_aid
2023-07-18 16:10:39,113 INFO     Search time: 2.4e+02
2023-07-18 16:10:39,113 INFO     Memory used: 0.1946 MB 7.56e+03 MB
2023-07-18 16:10:39,116 INFO     Plan length: 309
```

Figure 32: Execution with MBA + Random of problem 3*

problem3.pddl.soln

```
(fill_box robot2 box3 food depot)
(move_with_carrier robot1 carrier2 depot l7)
(load_box_on_carrier robot2 box3 food carrier1 depot zero one)
(move_with_carrier robot2 carrier1 depot l6)
(deliver_content robot2 box3 food carrier1 p6 l6)
(move_with_carrier robot1 carrier2 l7 l3)
(move_with_carrier robot2 carrier1 l6 l5)
(move_with_carrier robot1 carrier2 l3 l5)
(move_with_carrier robot1 carrier1 l5 l7)
(move_with_carrier robot2 carrier2 l5 depot)
(fill_box robot2 box4 tools depot)
(move_with_carrier robot1 carrier1 l7 l6)
(fill_box robot2 box1 medicine depot)
(move_with_carrier robot1 carrier1 l6 l3)
(load_box_on_carrier robot2 box1 medicine carrier2 depot zero one)
(fill_box robot2 box2 food depot)
(move_with_carrier robot1 carrier1 l3 depot)
(move_with_carrier robot2 carrier1 depot l3)
(load_box_on_carrier robot1 box4 tools carrier2 depot one two)
(move_with_carrier robot1 carrier2 depot l3)
(move_with_carrier robot1 carrier1 l3 depot)
(move_with_carrier robot2 carrier2 l3 l8)
(deliver_content robot2 box1 medicine carrier2 p8 l8)
(move_with_carrier robot1 carrier1 depot l7)
(move_with_carrier robot2 carrier2 l8 depot)
(move_with_carrier robot1 carrier1 l7 l2)
(unload_box_from_carrier robot2 box1 carrier2 depot two one)
(fill_box robot2 box1 medicine depot)
(move_with_carrier robot1 carrier1 l2 l3)
(load_box_on_carrier robot2 box2 food carrier2 depot one two)
(move_with_carrier robot2 carrier2 depot l5)
(move_with_carrier robot1 carrier1 l3 depot)
(move_with_carrier robot2 carrier2 l5 l8)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(load_box_on_carrier robot1 box1 medicine carrier1 depot zero one)
(move_with_carrier robot2 carrier2 l8 l6)
(move_with_carrier robot1 carrier1 depot l6)
(move_with_carrier robot1 carrier1 l6 l5)
(deliver_content robot1 box2 food carrier2 p5 l5)
(deliver_content robot1 box4 tools carrier2 p5 l5)
(move_with_carrier robot2 carrier1 l6 l8)
(move_with_carrier robot1 carrier2 l5 depot)
(unload_box_from_carrier robot1 box2 carrier2 depot two one)
(move_with_carrier robot1 carrier2 depot l8)
(move_with_carrier robot2 carrier2 l8 depot)
(move_with_carrier robot1 carrier1 l8 l3)
(move_with_carrier robot2 carrier2 depot l7)
(deliver_content robot1 box1 medicine carrier1 p3 l3)
(move_with_carrier robot1 carrier1 l3 l5)
(move_with_carrier robot2 carrier2 l7 l4)
(move_with_carrier robot1 carrier1 l5 depot)
(fill_box robot1 box2 tools depot)
(fill_box robot1 box3 tools depot)
(load_box_on_carrier robot1 box2 tools carrier1 depot one two)
(move_with_carrier robot2 carrier2 l4 l5)
(move_with_carrier robot1 carrier1 depot l8)
(move_with_carrier robot2 carrier2 l5 l2)
(move_with_carrier robot1 carrier1 l8 l6)
(move_with_carrier robot2 carrier2 l2 l6)
(move_with_carrier robot1 carrier2 l6 depot)
(load_box_on_carrier robot1 box3 tools carrier2 depot one two)
(move_with_carrier robot2 carrier1 l6 depot)
(unload_box_from_carrier robot2 box1 carrier1 depot two one)
(unload_box_from_carrier robot2 box4 carrier2 depot two one)
(move_with_carrier robot1 carrier2 depot l2)
(fill_box robot2 box4 medicine depot)
(load_box_on_carrier robot2 box4 medicine carrier1 depot one two)
(move_with_carrier robot2 carrier1 depot l2)
(move_with_carrier robot1 carrier1 l2 l3)
(deliver_content robot1 box4 medicine carrier1 p3 l3)
(move_with_carrier robot1 carrier1 l3 l2)
(move_with_carrier robot1 carrier2 l2 l4)
(move_with_carrier robot2 carrier1 l2 l6)
(unload_box_from_carrier robot2 box4 carrier1 l6 two one)
(move_with_carrier robot2 carrier1 l6 l5)
(move_with_carrier robot1 carrier2 l4 depot)
(deliver_content robot2 box2 tools carrier1 p5 l5)
(fill_box robot1 box1 medicine depot)
(load_box_on_carrier robot1 box1 medicine carrier2 depot one two)
(move_with_carrier robot1 carrier2 depot l7)
(move_with_carrier robot2 carrier1 l5 l7)
(move_with_carrier robot1 carrier1 l7 l6)
(deliver_content robot2 box1 medicine carrier2 p7 l7)
(move_with_carrier robot2 carrier2 l7 l8)
(move_with_carrier robot1 carrier1 l6 depot)
(move_with_carrier robot2 carrier2 l8 depot)
(move_with_carrier robot2 carrier1 depot l4)
(unload_box_from_carrier robot1 box1 carrier2 depot two one)
(move_with_carrier robot1 carrier2 depot l3)
(move_with_carrier robot2 carrier1 l4 depot)
(unload_box_from_carrier robot2 box2 carrier1 depot one zero)
(move_with_carrier robot1 carrier2 l3 l6)
(fill_box robot2 box1 medicine depot)
(move_with_carrier robot1 carrier2 l6 depot)
(move_with_carrier robot2 carrier2 depot l6)
(fill_box robot1 box2 tools depot)
(move_with_carrier robot2 carrier2 l6 l7)
(load_box_on_carrier robot1 box1 medicine carrier1 depot zero one)
(move_with_carrier robot2 carrier2 l7 l2)
(move_with_carrier robot1 carrier1 depot l4)
(move_with_carrier robot2 carrier2 l2 l6)
(deliver_content robot2 box3 tools carrier2 p6 l6)
(deliver_content robot1 box1 medicine carrier1 p4 l4)
(move_with_carrier robot2 carrier2 l6 l8)
(move_with_carrier robot1 carrier1 l4 depot)
(unload_box_from_carrier robot1 box1 carrier1 depot one zero)
(move_with_carrier robot2 carrier2 l8 l5)
(load_box_on_carrier robot1 box2 tools carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l8)
(move_with_carrier robot2 carrier2 l5 l8)
(move_with_carrier robot1 carrier2 l8 l6)
(deliver_content robot2 box2 tools carrier1 p8 l8)
(move_with_carrier robot1 carrier2 l6 depot)
(move_with_carrier robot2 carrier1 l8 l6)
(fill_box robot1 box1 tools depot)
(unload_box_from_carrier robot1 box3 carrier2 depot one zero)
(move_with_carrier robot2 carrier1 l6 depot)
(move_with_carrier robot2 carrier2 depot l3)
(unload_box_from_carrier robot1 box2 carrier1 depot one zero)
(load_box_on_carrier robot1 box1 tools carrier1 depot zero one)
(fill_box robot1 box3 tools depot)
(fill_box robot1 box2 medicine depot)
(move_with_carrier robot1 carrier1 depot l5)
(move_with_carrier robot2 carrier2 l3 depot)
(move_with_carrier robot1 carrier1 l5 l7)
(load_box_on_carrier robot2 box2 medicine carrier2 depot zero one)
(move_with_carrier robot1 carrier1 l7 depot)
(move_with_carrier robot2 carrier2 depot l2)
(load_box_on_carrier robot1 box3 tools carrier1 depot one two)
(move_with_carrier robot2 carrier2 l2 l7)
(move_with_carrier robot1 carrier1 depot l7)
(deliver_content robot2 box2 medicine carrier2 p7 l7)
(deliver_content robot2 box1 tools carrier1 p7 l7)
(move_with_carrier robot1 carrier1 l7 l5)
(move_with_carrier robot2 carrier1 l6 l7)
(move_with_carrier robot2 carrier2 l7 l6)
(move_with_carrier robot1 carrier1 l5 depot)
(unload_box_from_carrier robot1 box1 carrier2 depot one zero)
(move_with_carrier robot2 carrier2 l6 l8)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(fill_box robot1 box1 food depot)
(deliver_content robot2 box3 tools carrier1 p6 l6)
(load_box_on_carrier robot1 box1 tools carrier2 depot zero one)
(move_with_carrier robot1 carrier2 depot l8)
(move_with_carrier robot2 carrier1 l7 l5)
(move_with_carrier robot1 carrier2 l7 depot)
(move_with_carrier robot2 carrier1 depot l7)
(unload_box_from_carrier robot1 box1 carrier2 depot one zero)
(fill_box robot2 box3 medicine depot)
(move_with_carrier robot1 carrier2 l7 l6)
(move_with_carrier robot2 carrier1 l8 l7)
(move_with_carrier robot1 carrier1 depot l3)
(deliver_content robot2 box1 food carrier2 p1 l2)
(move_with_carrier robot2 carrier2 l2 depot)
(move_with_carrier robot1 carrier1 l3 l4)
(move_with_carrier robot2 carrier2 depot l3)
(deliver_content robot2 box3 medicine carrier2 p3 l3)
(move_with_carrier robot1 carrier1 l4 l3)
(move_with_carrier robot2 carrier2 l3 depot)
(unload_box_from_carrier robot2 box3 carrier2 depot two one)
(move_with_carrier robot1 carrier1 l3 l5)
(unload_box_from_carrier robot2 box1 carrier2 depot one zero)
(move_with_carrier robot1 carrier1 l5 depot)
(move_with_carrier robot2 carrier1 depot l7)
(load_box_on_carrier robot1 box2 medicine carrier2 depot zero one)
(move_with_carrier robot2 carrier1 l7 l2)
(fill_box robot1 box1 food depot)
(move_with_carrier robot1 carrier2 depot l6)
(move_with_carrier robot2 carrier1 l2 l6)
(move_with_carrier robot1 carrier1 l6 l2)
(deliver_content robot2 box2 medicine carrier2 p6 l6)
(move_with_carrier robot2 carrier2 l6 depot)
(load_box_on_carrier robot2 box1 food carrier2 depot one two)
(move_with_carrier robot1 carrier1 l2 l3)
(unload_box_from_carrier robot2 box2 carrier2 depot two one)
(move_with_carrier robot1 carrier1 l3 l4)
(move_with_carrier robot2 carrier2 depot l6)
(deliver_content robot2 box1 food carrier2 p6 l6)
(move_with_carrier robot1 carrier1 l4 depot)
(move_with_carrier robot2 carrier2 depot l6)
(move_with_carrier robot1 carrier2 l6 depot)
(fill_box robot2 box2 medicine depot)
(move_with_carrier robot2 carrier1 depot l8)
(unload_box_from_carrier robot1 box1 carrier2 depot one zero)
(load_box_on_carrier robot1 box2 medicine carrier2 depot zero one)
(move_with_carrier robot2 carrier1 l8 l2)
(move_with_carrier robot1 carrier2 depot l3)
(move_with_carrier robot2 carrier1 l2 l3)
(move_with_carrier robot1 carrier2 l3 l4)
(deliver_content robot1 box2 medicine carrier2 p4 l4)
(move_with_carrier robot2 carrier1 l3 l5)
(move_with_carrier robot1 carrier2 l4 l2)
(move_with_carrier robot2 carrier1 l5 l2)
(move_with_carrier robot1 carrier2 l2 depot)
(fill_box robot1 box1 tools depot)
(fill_box robot1 box3 food depot)
(move_with_carrier robot2 carrier1 l2 l6)
(unload_box_from_carrier robot1 box2 carrier2 depot one zero)
(move_with_carrier robot1 carrier2 depot l3)
(move_with_carrier robot2 carrier1 l6 depot)
(load_box_on_carrier robot2 box3 food carrier1 depot zero one)
(move_with_carrier robot1 carrier2 l3 l2)
(move_with_carrier robot2 carrier1 depot l4)
(move_with_carrier robot1 carrier2 l2 l4)
(deliver_content robot2 box3 food carrier1 p4 l4)
(move_with_carrier robot1 carrier1 l4 depot)
(unload_box_from_carrier robot1 box3 carrier1 depot one zero)
(move_with_carrier robot2 carrier2 l4 l3)
(fill_box robot1 box2 medicine depot)
(fill_box robot1 box3 food depot)
(load_box_on_carrier robot1 box2 medicine carrier1 depot zero one)
(move_with_carrier robot1 carrier1 depot l6)
(move_with_carrier robot2 carrier2 l3 l5)
(move_with_carrier robot1 carrier1 l6 l2)
(move_with_carrier robot2 carrier2 l5 l7)
(deliver_content robot1 box2 medicine carrier1 p2 l2)
(move_with_carrier robot2 carrier2 l7 l8)
(move_with_carrier robot1 carrier1 l2 l3)
(move_with_carrier robot2 carrier2 l8 l3)
(move_with_carrier robot1 carrier2 l3 l2)
(move_with_carrier robot2 carrier1 l3 depot)
(load_box_on_carrier robot2 box1 tools carrier1 depot one two)
(move_with_carrier robot2 carrier1 depot l2)
(move_with_carrier robot2 carrier2 l2 l4)
(deliver_content robot1 box1 tools carrier1 p1 l2)
```

Figure 33: MBA + Random generated plan of problem 3

TASK 3

3.1

DESCRIPTION

The strategy that led to the resolution of this task was divided into two parts.

Initially, the domain was implemented using PDDL 2.1 to transform actions into *durative actions*. The structure of the domain remained virtually unchanged from Task 1: no new actions were added, but the capacity modelling was modified, introducing fluents.

In this case, the following functions have been added:

- **(capacity ?c - carrier)**: indicates the current capacity of the carrier. In the problem file it is initialised to zero and its value depends on the number of boxes on the carrier.

- **(max_capacity)**: indicates the maximum capacity of the conveyor. In the problem file, it is initialised at three.

- **(content_weight ?c - content)**: indicates the weight of each content. In the problem file it is initialised with the different content values as required by the track.

- **(total_weight)**: Indicates the total weight of the transporter. It is initialised at zero and its value changes depending on the content added on the conveyor. It is also used to determine the duration of the robot's journey with the vehicle.

Plan Generation

The plan was generated with ***Temporal Fast Downward***, using the command:

planutils run tfd domain_da.pddl problem_da.pddl

within directory 3.1, contained in the Task 3 directory, which contains *domain_da.pddl* and *problem_da.pddl*.

The output generated by the planner (in the format: *<time>: (<action>) [<duration>]*) is as follows:

```
0.000000: (fill_box robot box5 drug loc1 p1) [0.00100000].
0.010000: (load_box_on_carrier robot box5 drug transporter loc1) [0.00200000].
0.020000: (fill_box robot box4 food loc1 p1) [0.00100000].
0.030000: (load_box_on_carrier robot box4 food transporter loc1) [0.00200000] [0.00200000].
0.00100000: (move_with_carrier robot box1 transporter loc1 loc2) [2.000000].
2.050000: (deliver_content robot box5 drug transporter p1 loc2) [0.00300000].
2.01100000: (move_with_carrier robot box1 transporter loc2 loc1) [1.000000].
3.070000: (unload_box_from_carrier robot box5 transporter loc1) [0.00200000].
3.080000: (fill_box robot box5 food loc1 p3) [0.00100000].
3.090000: (load_box_on_carrier robot box5 food transporter loc1) [0.00200000] [0.00200000].
3.10000000: (fill_box robot box3 drug loc1 p2) [0.00100000].
3.110000: (load_box_on_carrier robot box3 drug transporter loc1) [0.00200000] [0.00200000].
3.02100000: (move_with_carrier robot box1 transporter loc1 loc2) [3.000000].
6.03100000: (move_with_carrier robot box1 transporter loc2 loc3) [3.000000] [3.000000].
9.140000: (deliver_content robot box4 food transporter p3 loc3) [0.00300000] [0.00300000].
9.04100000: (move_with_carrier robot box1 transporter loc3 loc2) [1.000000].
10.160000: (deliver_content robot box5 food transporter p1 loc2) [0.00300000] [0.00300000].
10.170000: (deliver_content robot box3 drug transporter p2 loc2) [0.00300000].
```

3.2

The directory *plansys2_emergency, contained in* Task 3, contains the files that will be used to generate the plan.
The directory *pddl* contains the domain file written in task 3.1.
The *src* directory contains a file written in *C++* for each action in the domain.
The file *plan.txt* contains the output generated in the previous task.
The commands.*txt* file contains the Plansys2 terminal commands to create the problem instance. The last line of the file also contains the command *run plan-file <plan.txt>.*

To see the progress of action execution, two terminals must be opened and commands entered:

terminal 1:
cd plansys2_emergency
colcon build --symlink-install
source install/local_setup.bash
ros2 launch plansys2_emergency plansys2_emergency_launch.py

terminal 2:
cd plansys2_emergency
ros2 run plansys2_terminal plansys2_terminal
source commands.txt

The previously generated plan will be printed and, each time, the actions with a percentage that is increased by the scrolling of the action itself.

```
aiguy@ubu22:~/Desktop/plansys2_emergency$ ros2 run plansys2_terminal plansys2_terminal
[INFO] [1689662511.446883044] [terminal]: No problem file specified.
ROS2 Planning System console. Type "quit" to finish
> source commands.txt
done
done
done
done
done
done
The plan read from "plan.txt" is
0:      (fill_box robot box5 drug loc1 p1)      [0.001]
0.01:   (load_box_on_carrier robot box5 drug transporter loc1)  [0.002]
0.02:   (fill_box robot box4 food loc1 p1)      [0.001]
0.03:   (load_box_on_carrier robot box4 food transporter loc1)  [0.002]
0.001:  (move_with_carrier robot box1 transporter loc1 loc2)    [2]
2.05:   (deliver_content robot box5 drug transporter p1 loc2)   [0.003]
2.011:  (move_with_carrier robot box1 transporter loc2 loc1)    [1]
3.07:   (unload_box_from_carrier robot box5 transporter loc1)   [0.002]
3.08:   (fill_box robot box5 food loc1 p3)      [0.001]
3.09:   (load_box_on_carrier robot box5 food transporter loc1)  [0.002]
3.1:    (fill_box robot box3 drug loc1 p2)      [0.001]
3.11:   (load_box_on_carrier robot box3 drug transporter loc1)  [0.002]
3.021:  (move_with_carrier robot box1 transporter loc1 loc2)    [3]
6.031:  (move_with_carrier robot box1 transporter loc2 loc3)    [3]
9.14:   (deliver_content robot box4 food transporter p3 loc3)   [0.003]
9.041:  (move_with_carrier robot box1 transporter loc3 loc2)    [1]
10.16:  (deliver_content robot box5 food transporter p1 loc2)   [0.003]
10.17:  (deliver_content robot box3 drug transporter p2 loc2)   [0.003]
[(move_with_carrier robot box1 transporter loc2 loc1) 30%]
```

Figure 34: print action move_with_carrier

```
aiguy@ubu22:~/Desktop/plansys2_emergency$ ros2 run plansys2_terminal plansys2_terminal
[INFO] [1689662511.446883044] [terminal]: No problem file specified.
ROS2 Planning System console. Type "quit" to finish
> source commands.txt
done
done
done
done
done
done
The plan read from "plan.txt" is
0:      (fill_box robot box5 drug loc1 p1)      [0.001]
0.01:   (load_box_on_carrier robot box5 drug transporter loc1)  [0.002]
0.02:   (fill_box robot box4 food loc1 p1)      [0.001]
0.03:   (load_box_on_carrier robot box4 food transporter loc1)  [0.002]
0.001:  (move_with_carrier robot box1 transporter loc1 loc2)    [2]
2.05:   (deliver_content robot box5 drug transporter p1 loc2)   [0.003]
2.011:  (move_with_carrier robot box1 transporter loc2 loc1)    [1]
3.07:   (unload_box_from_carrier robot box5 transporter loc1)   [0.002]
3.08:   (fill_box robot box5 food loc1 p3)      [0.001]
3.09:   (load_box_on_carrier robot box5 food transporter loc1)  [0.002]
3.1:    (fill_box robot box3 drug loc1 p2)      [0.001]
3.11:   (load_box_on_carrier robot box3 drug transporter loc1)  [0.002]
3.021:  (move_with_carrier robot box1 transporter loc1 loc2)    [3]
6.031:  (move_with_carrier robot box1 transporter loc2 loc3)    [3]
9.14:   (deliver_content robot box4 food transporter p3 loc3)   [0.003]
9.041:  (move_with_carrier robot box1 transporter loc3 loc2)    [1]
10.16:  (deliver_content robot box5 food transporter p1 loc2)   [0.003]
10.17:  (deliver_content robot box3 drug transporter p2 loc2)   [0.003]
[(unload_box_from_carrier robot box5 transporter loc1) 30%]
```

Figure 35: Print action unload_box_from_carrier

```
aiguy@ubu22:~/Desktop/plansys2_emergency$ ros2 run plansys2_terminal plansys2_terminal
[INFO] [1689662511.446883044] [terminal]: No problem file specified.
ROS2 Planning System console. Type "quit" to finish
> source commands.txt
done
done
done
done
done
done
The plan read from "plan.txt" is
0:      (fill_box robot box5 drug loc1 p1)      [0.001]
0.01:   (load_box_on_carrier robot box5 drug transporter loc1)  [0.002]
0.02:   (fill_box robot box4 food loc1 p1)      [0.001]
0.03:   (load_box_on_carrier robot box4 food transporter loc1)  [0.002]
0.001:  (move_with_carrier robot box1 transporter loc1 loc2)    [2]
2.05:   (deliver_content robot box5 drug transporter p1 loc2)   [0.003]
2.011:  (move_with_carrier robot box1 transporter loc2 loc1)    [1]
3.07:   (unload_box_from_carrier robot box5 transporter loc1)   [0.002]
3.08:   (fill_box robot box5 food loc1 p3)      [0.001]
3.09:   (load_box_on_carrier robot box5 food transporter loc1)  [0.002]
3.1:    (fill_box robot box3 drug loc1 p2)      [0.001]
3.11:   (load_box_on_carrier robot box3 drug transporter loc1)  [0.002]
3.021:  (move_with_carrier robot box1 transporter loc1 loc2)    [3]
6.031:  (move_with_carrier robot box1 transporter loc2 loc3)    [3]
9.14:   (deliver_content robot box4 food transporter p3 loc3)   [0.003]
9.041:  (move_with_carrier robot box1 transporter loc3 loc2)    [1]
10.16:  (deliver_content robot box5 food transporter p1 loc2)   [0.003]
10.17:  (deliver_content robot box3 drug transporter p2 loc2)   [0.003]
[INFO] [1689662731.403764235] [executor_client]: Plan Succeeded

Successful finished
>
```

Figure 36: Completed plan printout

References

[1] Adam Green and other contributors 'PDDL 1.2'. https://planning.wiki/ref/pddl

[2] Artificial Intelligence Group - University of Basel 'Pyperplan: a lightweight STRIPS planner written in Python'. https://github.com/aibasel/pyperplan