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

Corso di Laurea Magistrale in Ingegneria Informatica

Relazione

Assignment del Corso di Intelligenza Artificiale

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

Anno Accademico 2022/2023

Indice

TASK 1

PDDL (**P**lanning **D**omain **D**efinition **L**anguage) è un linguaggio di descrizione del dominio di pianificazione per la definizione e la modellazione di problemi di pianificazione, creato con l'obiettivo di porsi come standard per permettere agli sviluppatori di comunicare in modo uniforme e interoperabile.

Il focus del team, nel contesto del primo task del progetto, è stato l'utilizzo della versione 1.2 di PDDL [1]. Questa versione, basata sui concetti già definiti per STRIPS, venne usata nel concorso AIPS 1998. Essa offre un insieme di costrutti chiari e ben definiti che consentono di specificare il dominio del problema, le azioni possibili e gli obiettivi da raggiungere.

Nel prosieguo della relazione verrà esaminato in dettaglio l'utilizzo di PDDL 1.2, analizzando la modellazione del dominio, le considerazioni implementative e i risultati ottenuti.

DESCRIZIONE DEL DOMINIO

Si consideri una problematica di natura logistica dei servizi di emergenza, in cui un certo numero di individui feriti si trova in posizioni fisse e non in movimento. Lo scopo della pianificazione consiste nell'organizzare le attività di uno o più agenti robotici al fine di consegnare a ciascuna persona delle scatole contenenti materiale di emergenza. Le ipotesi sono riportate nella traccia.

Il file del dominio (*domain.pddl*), contenuto nella directory Task 1, comprende:

Tipi

- **Location**: rappresenta la posizione di una scatola, un agente, un contenuto, un trasportatore o una persona.
- **Agent:** rappresenta l'agente robotico.
- **Box:** rappresenta la scatola.
- **Content:** rappresenta il contenuto della scatola.
- **Carrier:** rappresenta il trasportatore.
- **Person:** rappresenta la persona.
- **Box_number:** rappresenta il numero di scatole.

Predicati

- **(at ?x - (either agent box content carrier person) ?l - location)** indica la posizione l di un'entità x.
- **(empty ?b - box)** indica che la scatola b è vuota.
- **(filled ?b - box ?c - content)** indica che la scatola b è stata riempita con il contenuto c.
- **(carrier_loaded ?c - carrier ?b - box)** indica che il trasportatore c è stato caricato con la scatola b.
- **(needs ?p - person ?c - content)** indica che la persona p ha bisogno del contenuto c.
- **(has ?p - person ?c - content)** indica che la persona p ha ottenuto il contenuto c.
- **(capacity ?c - carrier ?bn - box_number)** indica la capacità corrente del trasportatore, ovvero il numero di scatole trasportate.
- **(next ?c - carrier ?bn1 ?bn2 - box_number)** definisce un ordinamento tra i valori possibili di box_number, stabilendo quali sono i valori successivi e precedenti.

Azioni

1. **fill_box**: questa azione ha il compito di riempire una scatola vuota con un certo contenuto. I suoi parametri sono *?robot* (l'agente che esegue l'azione), *?box* (la scatola da riempire), *?content* (il contenuto da inserire nella scatola) e *?at* (la posizione dell'agente e della scatola). La precondizione richiede che l'agente, la scatola e il contenuto si trovino nella stessa posizione e che la scatola sia vuota. L'effetto dell'azione è di riempire la scatola con il contenuto indicato.

```
(: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**: questa azione si occupa di caricare una scatola su un trasportatore. I suoi parametri sono *?robot*, *?box*, *?content*, *?carrier* (il trasportatore su cui caricare la scatola), *?at* (la posizione dell'agente, della scatola e del trasportatore) e *?bnbefore* e *?bnafter* (servono a manipolare la capacità del trasportatore prima e dopo aver caricato la scatola). La precondizione richiede che l'agente, la scatola, il contenuto e il trasportatore si trovino nella stessa posizione, che la scatola sia piena e che il trasportatore abbia ancora spazio disponibile per caricare la scatola. L'effetto dell'azione è di caricare la scatola sul trasportatore, modificandone così la capacità corrente.

```
(: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**: questa azione ha il compito di scaricare una scatola da un mezzo di trasporto. I suoi parametri sono *?robot*, *?box*, *?carrier*, *?at* e ?*bnbefore* e ?*bnafter*. La precondizione richiede che l'agente, la scatola, il trasportatore si trovino nella stessa posizione e che la scatola sia vuota. L'effetto dell'azione è di scaricare la scatola dal trasportatore, modificandone così la capacità corrente.

```
(: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**: questa azione ha il compito di spostare l'agente e il mezzo di trasporto da una posizione all'altra. I suoi parametri sono *?robot*, *?carrier*, ?*from* (la posizione di partenza), ?*to* (la posizione di arrivo). La precondizione richiede che l'agente e il mezzo di trasporto si trovino nella posizione di partenza. L'effetto dell'azione è di spostare l'agente e il mezzo di trasporto nella posizione di arrivo.

```
(: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**: questa azione ha il compito di consegnare il contenuto di una scatola a una persona che ne ha bisogno. I suoi parametri sono *?robot*, *?box*, *?content*, *?carrier*, *?person* (la persona a cui consegnare il contenuto) e *?at* (la posizione dell'agente, della scatola, del trasportatore e della persona). La precondizione richiede che l'agente, la scatola, il contenuto, il mezzo di trasporto e la persona si trovino nella stessa posizione e che la scatola sia piena. L'effetto dell'azione è di consegnare il contenuto della scatola alla persona indicata, svuotando la scatola.

```
  (: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))
  )

)
```

SCELTE PROGETTUALI

Data l'impossibilità di utilizzare i *fluents* e le *functions* per rappresentare valori numerici, si è scelto di optare per una soluzione alternativa, usando solo costrutti della versione 1.2 di PDDL. Per poter contare quante scatole ci fossero sul carrello e tenere traccia della capacità del carrello è stato usato il predicato **next**. Questo ha permesso di incrementare il valore di **capacity**. Ad esempio, nel file del primo problema, capacity ha come valori ammissibili *zero*, *one*, *two, three* e *four:* ciò evita che capacity possa superare il limite massimo imposto dalla traccia.

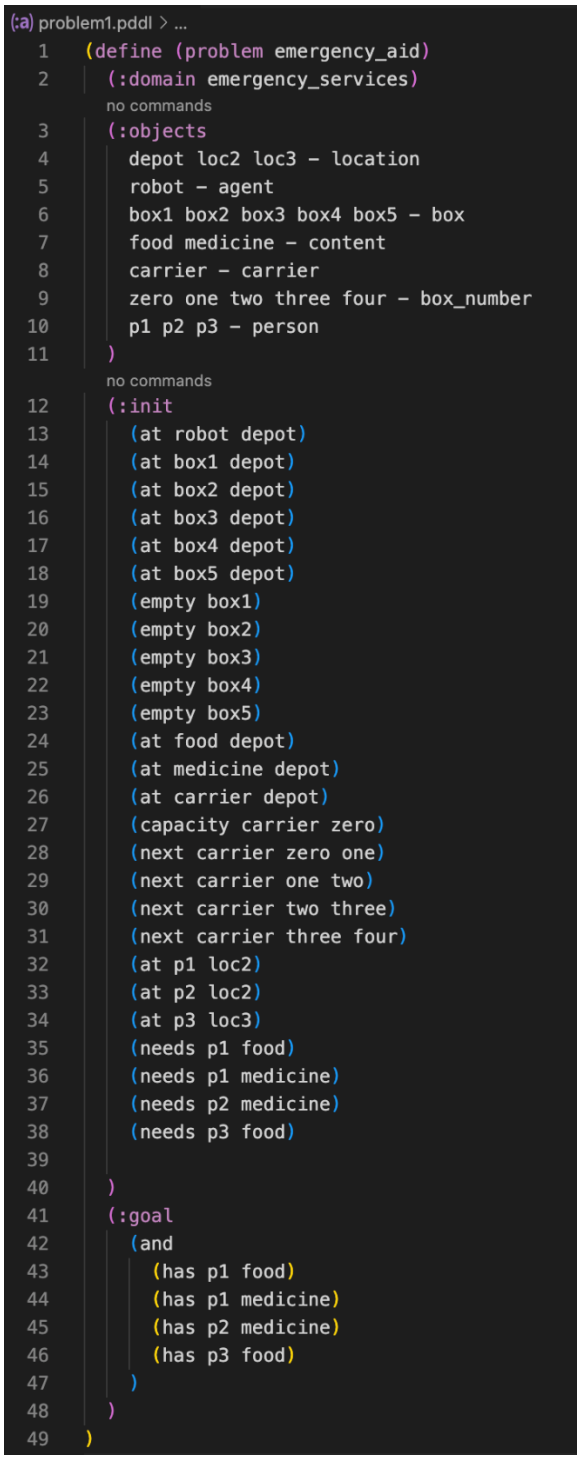
```
(next carrier zero one)
(next carrier one two)
(next carrier two three)
(next carrier three four)
```

Il valore di capacity viene incrementato come effetto dell'azione *load_box_on_carrier* e decrementato dopo l'azione *unload_box_from_carrier.*

È stata fatta l'assunzione che il robot si muova solo con il carrello. L'azione che descrive il movimento dell'agente (**move**) è stata commentata poiché non viene mai utilizzata dal robot, il quale piuttosto si sposta con il carrello per trasportare i contenuti che servono ai feriti.

TASK 2

All'interno della directory Task 2, è contenuta la cartella Problems, che comprende le tre istanze dei problemi richiesti dalla traccia, chiamati rispettivamente *problem1*, *problem2* e *problem3*.



```
(:a) 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)
    )
  )
)
```

Figura 1: file problem1.pddl

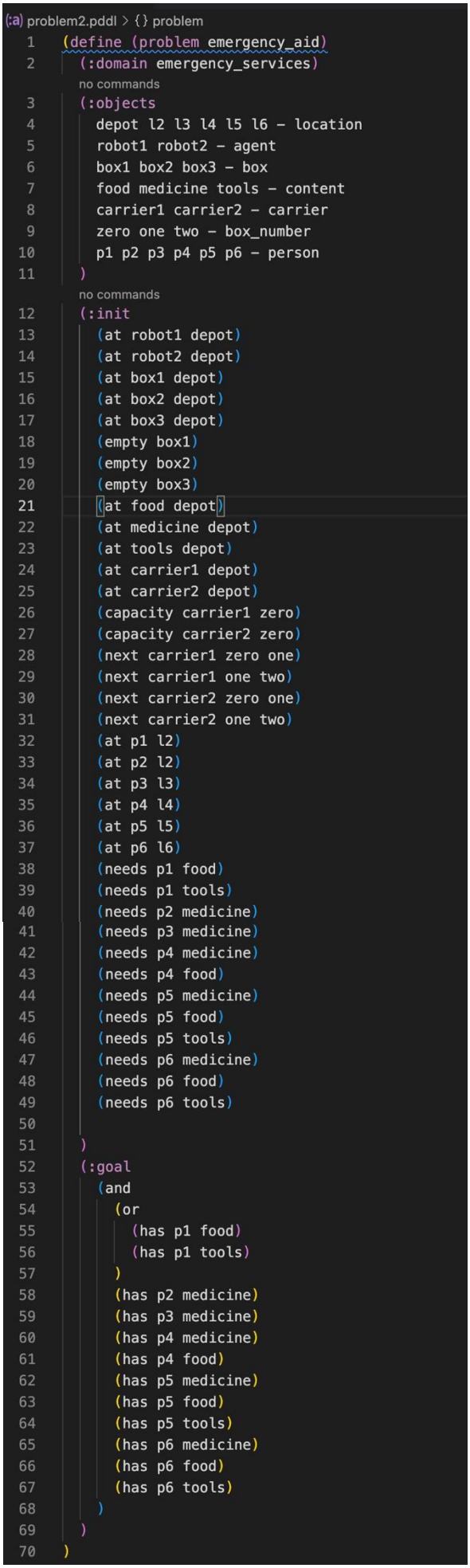


Figura 2: file problem2.pddl

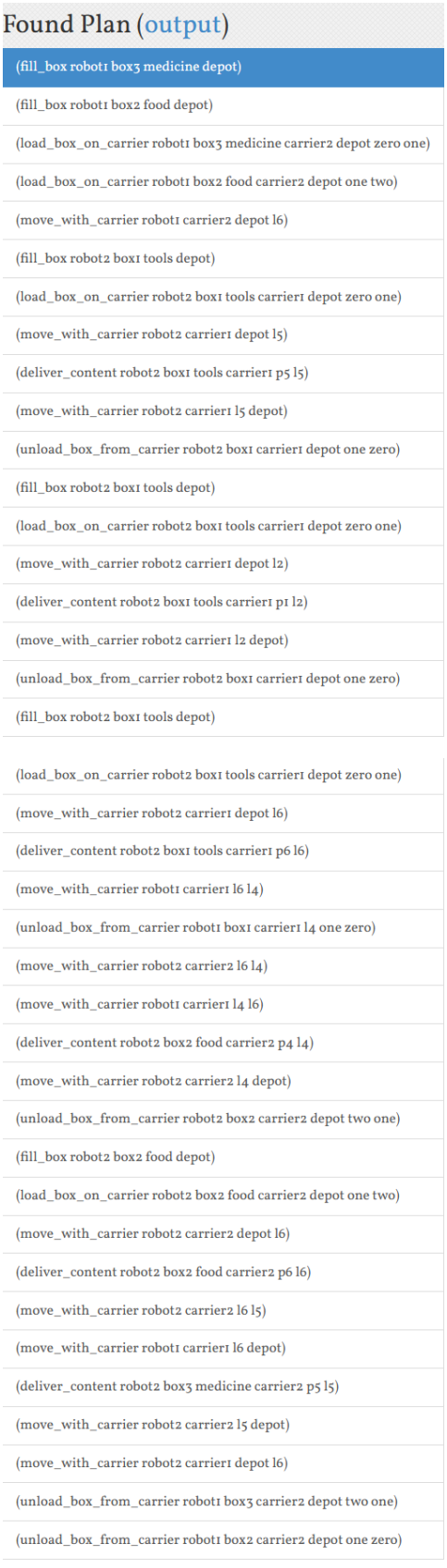
problem3.pddl > ...

```
(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)
    )
  )
)
```

Figura 3: file problem3.pddl

Nella stesura dei problemi, si è tenuto conto dei requisiti di progettazione richiesti dalla traccia, pertanto è stato inserito nella sezione dedicata al goal il costrutto “or”. Tuttavia, poiché sia PDDL4J sia il planner adottato (Pyperplan) non sono in grado di gestire la disgiunzione tra i predicati, si è deciso di rilassare il problema e di porre tutte gli obiettivi del problema due e tre in and. Per completezza, si riporta comunque un piano generato da PDDL Editor per mostrare il supporto all’or e la generazione di un piano tenendo in considerazione i requisiti del problema originale.



(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)

Figura 4: Piano generato da PDDL Editor con problem2.pddl

Come precedentemente accennato, per la realizzazione di questo task, è stato utilizzato il planner Pyperplan [2], un pianificatore leggero STRIPS scritto in Python. Questo pianificatore comprende un'ampia gamma di algoritmi di ricerca, tra cui:

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

Per quanto riguarda le euristiche, Pyperplan mette a disposizione le seguenti opzioni:

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

Un aspetto notevole di Pyperplan è la sua natura altamente estensibile e modificabile, consentendo lo sviluppo di un ulteriore algoritmo di ricerca e altre euristiche, non presenti nel codice sorgente originale del planner.

Questo planner è contenuto nella cartella *Task2/pyperplain-main*. All'interno di questa cartella, nel percorso *pyperplan-main\pyperplan\search\astar.py* si trova l'algoritmo di ricerca e all'interno di *pyperplan-main\pyperplan\heuristics\relaxation.py* si trovano le euristiche implementate.

ALGORITMO DI RICERCA

L'algoritmo di ricerca implementato è ***Memory Bounded A**** (MBA*).

L'utilizzo di questo algoritmo è risultato vantaggioso vista la complessità dei problemi e il numero elevato di nodi generati. Quest'algoritmo eredita tutte le caratteristiche di A*, già presente all'interno del planner ma con particolare attenzione alla memoria.

Viene utilizzato un heap *open* per mantenere i nodi ancora da esplorare, ordinati in base al costo totale stimato (f) per raggiungere l'obiettivo. Viene eseguito un ciclo principale che continua finché ci sono nodi in *open*. Durante ogni iterazione, viene estratto il nodo con il costo totale stimato minore, vengono esplorati i suoi successori e vengono aggiunti in *open* se non sono stati ancora visitati o se è stata trovata una strada più economica per raggiungerli. A ogni iterazione, un parametro intero *counter* viene incrementato per contare il numero di valori inseriti nella coda. Ogni tot valori (in questo caso mille), avviene un check sulla memoria: se la dimensione della coda supera il limite massimo di memoria (*max_memory*) viene eseguita la "potatura" della coda, mantenendo solo i nodi più piccoli. Il vantaggio è evidente: ridurre la quantità di memoria richiesta per conservare la coda di priorità.

La sezione modificata di A* che permette di controllare la dimensione della memoria è la seguente:

```
#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)
```

Segue il codice dell'algoritmo commentato.

```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
```

Figura 5: Algoritmo MBA Search*

EURISTICHE

RELAXED GOAL COUNT (RGC)

La prima euristica implementata si chiama **Relaxed Goal Count (RGC)** e stima il numero di goal che non sono ancora stati raggiunti nello stato corrente. L'euristica RGC non considera le distanze o i costi per raggiungere l'obiettivo, preferendo fornire una semplice stima basata sul conteggio dei goal rimanenti.

L'euristica è implementata mediante una classe che contiene due metodi: __init__ e __call__ . Il primo è un semplice costruttore mentre il secondo contiene il codice che realizza l'euristica vera e propria.

In __call__ viene calcolata la lista dei goal rimanenti confrontando i fatti nello stato del goal con i fatti nello stato del nodo corrente. Infine, il valore euristico h_value corrisponde al numero di goal rimanenti.

```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
```

Figura 6: classe hRGCHeuristic (presente nel file relaxation.py)

Per testare l'efficienza dell'algoritmo Memory Bounded A*, si è deciso di generare (utilizzando sempre la stessa euristica) il piano sullo stesso problema, prima con A* e poi con Memory Bounded A*.

Per stampare la memoria usata è stata utilizzato il modulo *Tracemalloc*; il primo valore visualizzato fa riferimento alla memoria corrente e il secondo al valore di picco della memoria (corrisponde al massimo utilizzo di memoria da quando è stato invocato l'algoritmo al termine della sua esecuzione).

```
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
```

Figura 7: esecuzione con MBA+ RGC del problema 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
```

Figura 8: esecuzione con A + RGC del problema 1*

Come si può notare, la quantità di memoria utilizzata dai due algoritmi cambia notevolmente (43.28 MB di Memory Bounded A* contro 273.4 MB di A*), così come i tempi di esecuzione (0.89 secondi di Memory Bounded A* contro 7.1 secondi di A*).

Nonostante i vantaggi di MBA*, tuttavia, A* si mostra in grado di trovare un piano più corto.

problem1.pddl.soln

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

Figura 9: piano generato da A + RGC*

problem1.pddl.soln

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

Figura 10: Piano generato da MBA + RGC*

Seguono le esecuzioni del problema 2 e del problema 3 con MBA* e con i loro piani generati.

```
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
```

Figura 11: esecuzione con MBA + RGC del problema 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)
```

Figura 12: piano generato da MBA + RGC del problema 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
```

Figura 13: esecuzione con MBA + RGC sul problema 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)
```

Figura 14: piano generato da MBA + RGC sul problema 3*

HAMMING

Questa euristica stima il numero di differenze tra lo stato corrente e lo stato obiettivo calcolando la distanza di Hamming.

```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
```

Figura 15: classe hHammingHeuristic (presente nel file relaxation.py)

Nel metodo __call__, viene calcolato il valore dell'euristica, in questo modo:

- Viene ottenuto lo stato corrente del nodo e viene creato un insieme chiamato current_state contenente gli elementi dello stato corrente.
- Viene creato un insieme chiamato goal_state che contiene gli elementi dello stato obiettivo.
- Viene calcolato h_value come somma dei confronti tra gli elementi di current_state e goal_state. La funzione zip viene utilizzata per iterare contemporaneamente sugli elementi delle due liste.
- All'interno del loop, viene confrontato ogni elemento corrente con l'elemento obiettivo corrispondente. Se i due elementi sono diversi (cioè il confronto current_fact != goal_fact è vero), h_value viene incrementato.
- Alla fine del loop, h_value conterrà il numero di elementi che differiscono tra current_state e goal_state.

Anche in questo caso è stato generato il piano sia con A* che con MBA*. I risultati sono in linea con quelli precedentemente ottenuti. Come previsto, MBA* permette di ridurre il consumo di memoria e migliorare i tempi di ricerca, a discapito della lunghezza del piano.

```
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
```

Figura 16: esecuzione con MBA + Hamming sul problema 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)
```

Figura 17: piano generato MBA + Hamming sul problema 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
```

Figura 18: esecuzione con A + Hamming del problema 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)
```

Figura 19: piano generato da A + Hamming sul problema 1*

Anche per questa euristica è stato generato il piano sul problema 2. Confrontando i risultati ottenuti con il piano generato sullo stesso problema e con l'euristica Relaxed Goal Count, si possono notare delle differenze sulla memoria usata e sul tempo di esecuzione:

	MEMORIA	TEMPO DI RICERCA	LUNGHEZZA PIANO
HAMMING	293.1 MB	19 s	79
RGC	195.2 MB	14 s	74

L'euristica di Hamming calcola la distanza tra lo stato corrente e lo stato obiettivo contando il numero di posizioni diverse tra i due stati e ciò può essere molto oneroso in termini di calcolo.

```
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
```

Figura 20: esecuzione con MBA + Hamming del problema 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)
```

Figura 21: Piano generato da MBA + Hamming del problema 2*

Segue un confronto dell'euristica RGC e l'euristica Hamming sul problema 3.

	MEMORIA	TEMPO DI RICERCA	LUNGHEZZA PIANO
HAMMING	797.7 MB	54 s	119
RGC	615 MB	42 s	117

```
antoninovaccarella@Macbook-Pro-di-Antonino pyperplan-main % pyperplan -H hhamming -s mba domain.pddl problem3.pddl
2023-07-18 15:53:54,788 INFO     using search: mb_astar_search
2023-07-18 15:53:54,788 INFO     using heuristic: hHammingHeuristic
2023-07-18 15:53:54,789 INFO     Parsing Domain /Users/antoninovaccarella/Desktop/pyperplan-main/domain.pddl
2023-07-18 15:53:54,790 INFO     Parsing Problem /Users/antoninovaccarella/Desktop/pyperplan-main/problem3.pddl
2023-07-18 15:53:54,790 INFO     8 Predicates parsed
2023-07-18 15:53:54,790 INFO     5 Actions parsed
2023-07-18 15:53:54,790 INFO     30 Objects parsed
2023-07-18 15:53:54,790 INFO     0 Constants parsed
2023-07-18 15:53:54,790 INFO     Grounding start: emergency_aid
2023-07-18 15:53:54,827 INFO     Relevance analysis removed 0 facts
2023-07-18 15:53:54,828 INFO     Grounding end: emergency_aid
2023-07-18 15:53:54,828 INFO     200 Variables created
2023-07-18 15:53:54,828 INFO     3744 Operators created
2023-07-18 15:53:54,831 INFO     Search start: emergency_aid
2023-07-18 15:53:54,831 INFO     Initial h value: 18.000000
2023-07-18 15:53:55,268 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:53:55,699 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:53:56,132 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:54:48,392 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:54:48,717 INFO     Memory limit reached. Pruning open nodes.
2023-07-18 15:54:48,784 INFO     Goal reached. Start extraction of solution.
2023-07-18 15:54:48,784 INFO     139146 Nodes expanded
2023-07-18 15:54:49,218 INFO     Search end: emergency_aid
2023-07-18 15:54:49,218 INFO     Search time: 5.4e+01
2023-07-18 15:54:49,218 INFO     Memory used: 0.1928 MB 797.7 MB
2023-07-18 15:54:49,221 INFO     Plan length: 119
```

Figura 22: esecuzione con MBA + Hamming del problema 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)
```

Figura 23: Piano generato da MBA + Hamming sul problema 3*

RANDOM

Questa classe implementa l'euristica Random. È stata implementata poiché ha fornito una base di confronto per valutare l'efficienza delle altre euristiche.

```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
```

Figura 24: classe RandomHeuristic (presente nel file relaxation.py)

L'euristica non tiene conto delle caratteristiche dello stato corrente o dello stato obiettivo, ma restituisce semplicemente un valore casuale compreso tra 0 e 100. È stata utile per testare le euristiche precedenti e verificare il loro comportamento rispetto all'euristica random sulle stesse istanze dei problemi.

```
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
```

Figura 25: esecuzione con MBA + Random del problema 1 (h_value = 88)*

problem1.pddl.soln

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

*Figura 26: Piano generato da Memory Bounded A**

Come risulta dalle immagini sottostanti, il comportamento dell'algoritmo con quest'euristica è totalmente casuale e varia indipendentemente dal valore iniziale dell'euristica.

```
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
```

Figura 27: esecuzione con MBA + Random del problema 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
```

Figura 28: esecuzione con MBA + Random del problema 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
```

Figura 29: esecuzione con MBA + Random del problema 1 (h_value = 67)*

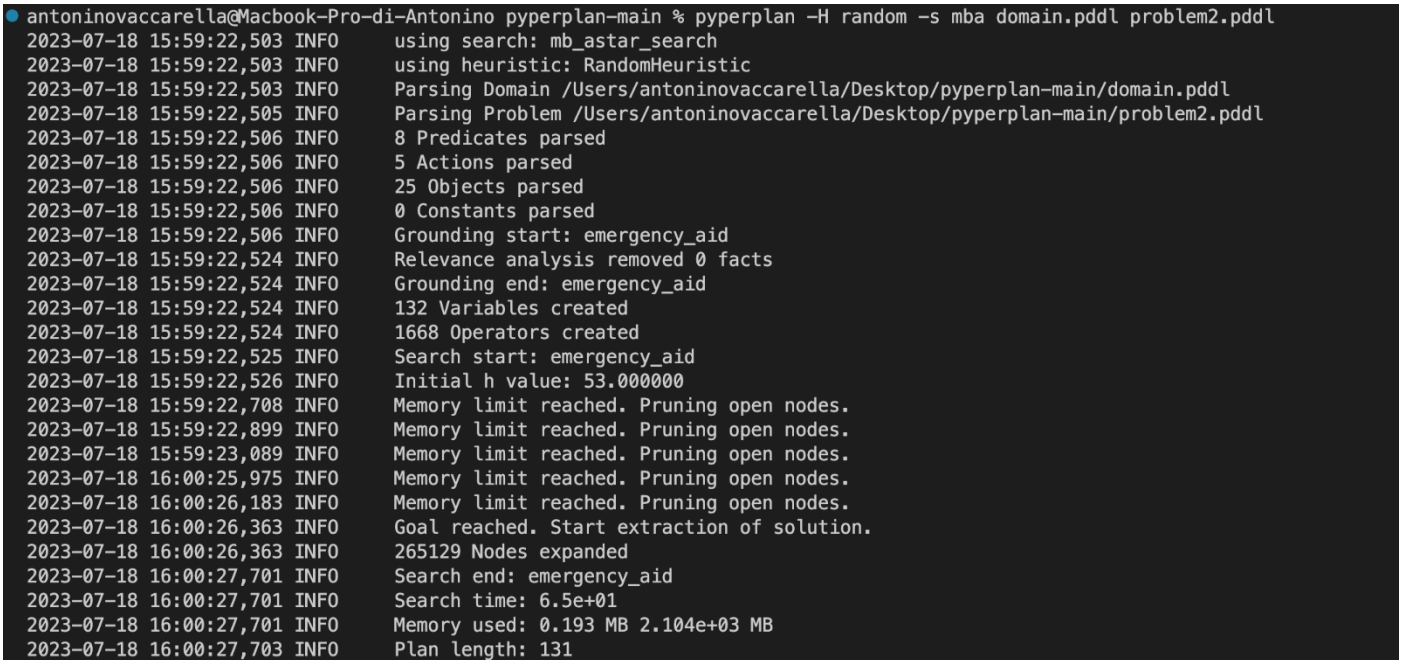
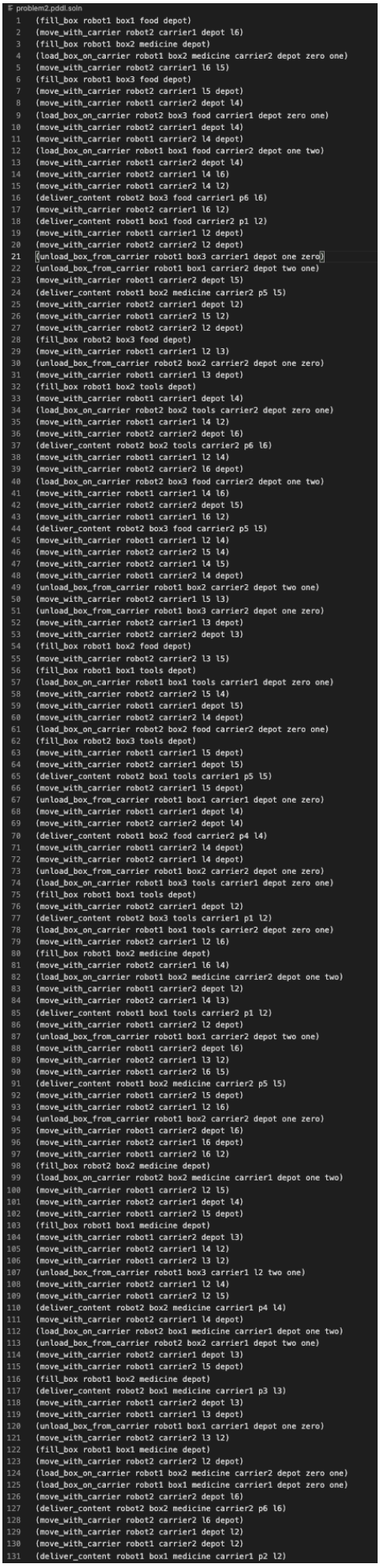


Figura 30: esecuzione con MBA + Random del problema 2*



problem2.pddl.soln

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

Figura 31: piano generato da MBA + Random con problema 2*

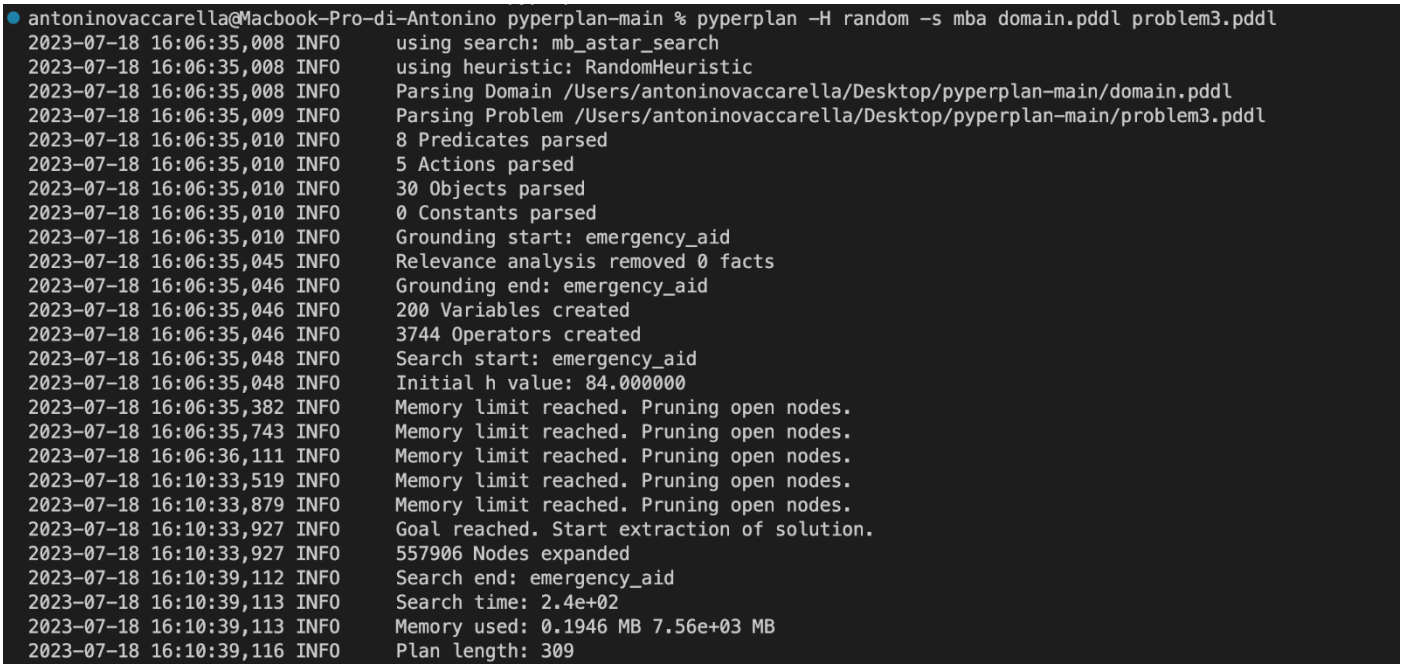


Figura 32: esecuzione con MBA + Random del problema 3*

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

Figura 33: piano generato da MBA + Random del problema 3

TASK 3

3.1

DESCRIZIONE

La strategia che ha portato alla risoluzione di questo task è stata suddivisa in due parti.

Inizialmente è stato realizzato il dominio usando PDDL 2.1 per trasformare le azioni in *durative action*. La struttura del dominio è rimasta pressoché invariata rispetto al Task 1: non sono state aggiunte nuove azioni ma è stata modificata la modellazione della capacità, introducendo i fluents.

In questo caso sono state aggiunte le seguenti functions:

- **(capacity ?c - carrier)**: indica la capacità corrente del trasportatore. Nel problem file viene inizializzata a zero e il suo valore dipende dal numero di scatole presenti sul mezzo di trasporto.

- **(max_capacity)**: indica la capacità massima del trasportatore. Nel problem file viene inizializzata a tre.

- **(content_weight ?c - content)**: indica il peso di ciascun content. Nel problem file viene inizializzato con i diversi valori dei content, come richiesto dalla traccia.

- **(total_weight)**: indica il peso totale del mezzo di trasporto. Viene inizializzato a zero e il suo valore cambia a seconda dei content aggiunti sul trasportatore. Serve a determinare anche la durata del viaggio del robot con il mezzo.

Generazione del piano

Il piano è stato generato con ***Temporal Fast Downward***, utilizzando il comando:

planutils run tfd domain_da.pddl problem_da.pddl

all'interno della directory 3.1, contenuta nella directory Task 3, che contiene *domain_da.pddl* e *problem_da.pddl*.

L'output generato dal planner (nel formato: *<time>: (<action>) [<duration>]*) è il seguente:

```
0.00000000: (fill_box robot box5 drug loc1 p1) [0.00100000]
0.01000000: (load_box_on_carrier robot box5 drug transporter loc1) [0.00200000]
0.02000000: (fill_box robot box4 food loc1 p1) [0.00100000]
0.03000000: (load_box_on_carrier robot box4 food transporter loc1) [0.00200000]
0.00100000: (move_with_carrier robot box1 transporter loc1 loc2) [2.00000000]
2.05000000: (deliver_content robot box5 drug transporter p1 loc2) [0.00300000]
2.01100000: (move_with_carrier robot box1 transporter loc2 loc1) [1.00000000]
3.07000000: (unload_box_from_carrier robot box5 transporter loc1) [0.00200000]
3.08000000: (fill_box robot box5 food loc1 p3) [0.00100000]
3.09000000: (load_box_on_carrier robot box5 food transporter loc1) [0.00200000]
3.10000000: (fill_box robot box3 drug loc1 p2) [0.00100000]
3.11000000: (load_box_on_carrier robot box3 drug transporter loc1) [0.00200000]
3.02100000: (move_with_carrier robot box1 transporter loc1 loc2) [3.00000000]
6.03100000: (move_with_carrier robot box1 transporter loc2 loc3) [3.00000000]
9.14000000: (deliver_content robot box4 food transporter p3 loc3) [0.00300000]
9.04100000: (move_with_carrier robot box1 transporter loc3 loc2) [1.00000000]
10.16000000: (deliver_content robot box5 food transporter p1 loc2) [0.00300000]
10.17000000: (deliver_content robot box3 drug transporter p2 loc2) [0.00300000]
```

3.2

Nella directory *plansys2_emergency,* contenuta in Task 3, sono contenuti i file che serviranno per la generazione del piano.
La directory *pddl* contiene il file del dominio scritto nel task 3.1.
La directory *src* contiene un file scritto in *C++* per ogni azione presente nel dominio.
Il file *plan.txt* contiene l'output generato nel task precedente.
Il file *commands.txt* contiene i comandi del terminale Plansys2 per creare l'istanza del problema. L'ultima riga del file contiene anche il comando *run plan-file <plan.txt>.*

Per vedere il progresso dell'esecuzione delle azioni, bisogna aprire due terminali e inserire i comandi:

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

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

Verrà stampato il piano generato precedentemente e, volta per volta, le azioni con una percentuale che viene incrementata dallo scorrimento dell'azione stessa.

```
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%]
```

Figura 34: stampa azione 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%]
```

Figura 35: stampa azione 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
>
```

Figura 36: stampa piano completata

Riferimenti

[1] Adam Green e altri collaboratori “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