

Relazione del progetto

di

# **Intelligenza Artificiale**

Corso di laurea magistrale

in

ingegneria informatica

A.A. 2022/2023

Bruno Guzzo (mat. 242504)

Marco Greco (mat. 242455)

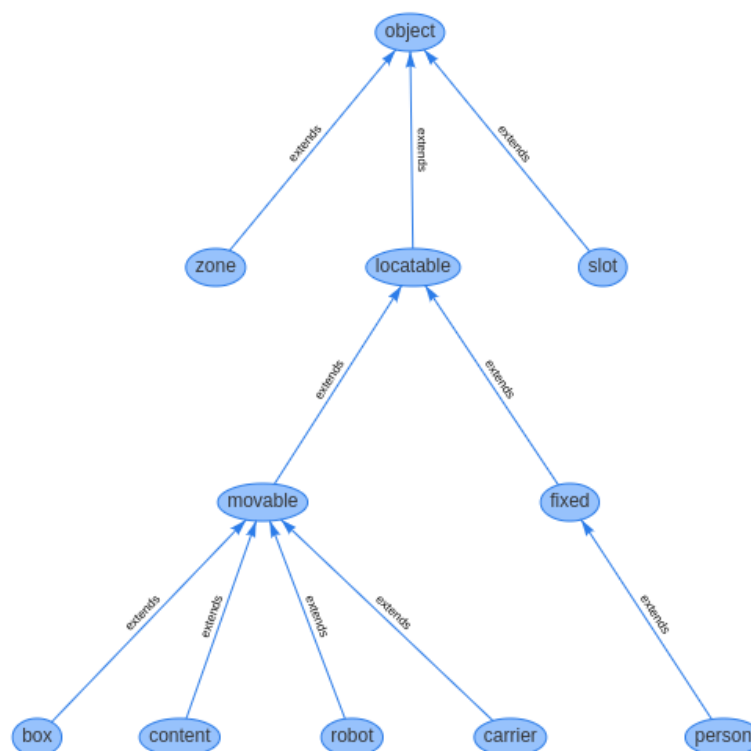
# Sommario

<b>Modelling</b>	<b>2</b>
Tipi	2
Predicati	4
Azioni	5
Carico di un contenuto in una scatola	5
Scarico del contenuto da una scatola	6
Carico di una scatola sul carrello	6
Scarico di una scatola dal carrello	7
Movimento del robot	8
Movimento del robot con carrello	8
Note sul dominio	8
<b>Classical planning</b>	<b>10</b>
Planner	10
Euristica	13
Instance 1	15
Instance 2	18
Instance 3 (Bonus)	22
Note aggiuntive sul planner usato	27
<b>Temporal Planning &amp; Robotics</b>	<b>29</b>
Temporal Planning	29
Robotics Planning	35
<b>Deliverables</b>	<b>51</b>

# Modelling

## Tipi

Per modellare il problema dato usando il linguaggio PDDL è necessario definire un dominio che tenga conto delle specifiche richieste. A tale scopo, si è deciso di usare una gerarchia dei tipi in modo da distinguere i vari componenti del sistema che interagiscono tra loro e definire le loro proprietà, tale gerarchia è riportata di seguito.



Per comprendere il ruolo dei tipi usati se ne riporta una breve descrizione per ognuno:

1. **Object:** Tipo padre degli altri tipi.
2. **Zone:** Rappresenta una zona del problema ovvero un luogo dove possono essere disposte persone, robot, carrelli e materiale.
3. **Locatable:** Tipo che rappresenta tutti gli oggetti che possono trovarsi in una posizione determinata, ovvero una zona.

4. **Slot:** Rappresenta uno scompartimento del carrello il quale può alloggiare una sola scatola per volta. Un carrello può avere molteplici compartimenti.
5. **Movable:** Tipo che rappresenta gli oggetti che possono muoversi nella specifica del problema.
6. **Fixed:** Tipo che rappresenta gli oggetti che non possono essere mossi nella specifica del problema.
7. **Box:** Rappresenta una scatola la quale può contenere solo un contenuto e può essere alloggiata in un solo compartimento del carrello.
8. **Content:** Rappresenta un contenuto, ovvero un oggetto che può essere inserito in una scatola e di cui le persone definite nel problema possono avere bisogno. Come assegnato nella traccia vi sono solo tre tipi di contenuto: cibo, strumenti e medicine i quali saranno definiti nelle istanze dei problemi in seguito come oggetti di tipo contenuto.
9. **Robot:** Rappresenta un robot, ovvero un'entità autonoma capace di muoversi, agganciare e spostare un carrello, riempire e svuotare una scatola per soddisfare il bisogno di una persona, inserire e rimuovere una scatola da un compartimento del carrello.
10. **Carrier:** Rappresenta un carrello, esso può essere agganciato ed essere spostato da un robot, possiede uno o più slot e può trovarsi in una determinata posizione.
11. **Person:** Rappresenta una persona la quale si trova in una determinata posizione e può necessitare di uno o più tipi di contenuto.

Questa gerarchia è stata definita nel file di dominio (*rescue\_domain.pddl*) mediante l'uso della dichiarazione *types* come riportato nel seguente frammento di codice.

```
(:types
  zone locatable slot - object
  movable fixed - locatable
  person - fixed
  box content robot carrier - movable
)
```

# Predicati

Dopo aver definito i tipi per modellare lo stato del sistema è necessario definire i predicati i quali saranno manipolati (affermati o negati) dalle azioni possibili al fine di raggiungere uno stato obiettivo (insieme di predicati da soddisfare). Dal momento che tali predicati devono specificare la configurazione del sistema dovranno rappresentare:

1. La locazione degli oggetti
2. Lo stato delle scatole: piene o vuote
3. Lo stato dei compartimenti del carrier: alloggia una scatola o meno
4. L'appartenenza di uno specifico compartimento al carrello
5. I bisogni delle persone
6. I contenuti posseduti dalle persone

I predicati sono stati definiti nel file di dominio mediante la dichiarazione *predicates* della quale si riporta il codice di seguito.

```
(:predicates
  (at ?obj - locatable ?zone - zone)
  (free-box ?box - box)
  (box-filled ?box - box ?cont - content)
  (box-on-carrier ?box - box ?car - carrier)
  (person-has-content ?per - person ?cont - content)
  (person-needs-content ?per - person ?cont - content)
  (free-slot ?slot - slot)
  (slot-of-carrier ?slot - slot ?car - carrier)
)
```

Analizziamo ora nello specifico ogni predicato definendone il significato:

1. **at**: Asserisce la posizione di un oggetto posizionabile in una zona.
2. **free-box**: Asserisce che una scatola sia libera o meno.
3. **box-filled**: Asserisce che una scatola sia piena e di quale contenuto.
4. **box-on-carrier**: Asserisce che una scatola sia caricata su un carrello (in un suo compartimento).
5. **person-has-content**: Asserisce che una persona possiede un contenuto.
6. **person-needs-content**: Asserisce la necessità di una persona di avere un contenuto.
7. **free-slot**: Asserisce che un compartimento è libero.
8. **slot-of-carrier**: Asserisce che un determinato slot appartiene ad un determinato carrello.

Tutti questi predicati verranno usati in fase di definizione delle istanze dei problemi assegnati (vedi sezione [Classical Planning](#)) per dichiarare le condizioni iniziali del sistema.

## Azioni

Le azioni possibili nel sistema, in accordo alla traccia, devono consentire che il robot possa:

1. Muoversi da una zona all'altra
2. Agganciare e spostare con esso un carrello
3. Caricare un contenuto in una scatola da una zona
4. Scaricare il contenuto da una scatola e soddisfare il bisogno di una persona
5. Caricare una scatola in un compartimento del carrello
6. Scaricare una scatola da un compartimento del carrello e lasciarla in una zona

Ognuna di queste azioni è stata definita nel file di dominio mediante la dichiarazione *action* nella quale vengono specificate le precondizioni (predicati che devono essere veri) e gli effetti (predicati da asserire o negare) in modo da portare il sistema in uno stato conforme all'azione applicata. Nelle successive sezioni si analizziamo gli effetti di ognuna di esse.

### Carico di un contenuto in una scatola

Il caricamento del contenuto in una scatola è applicabile se il robot, il contenuto e la scatola si trovano nella stessa zona e la scatola è vuota. L'effetto di questa azione sarà che la scatola non sarà più vuota ma conterrà un contenuto specifico.

```
(:action put-content-in-box
  :parameters (?box - box ?cont - content ?rob - robot ?zone - zone)
  :precondition (and
    (free-box ?box)
    (at ?cont ?zone)
    (at ?box ?zone)
    (at ?rob ?zone)
```

```

)
:effect (and
  (box-filled ?box ?cont)
  (not (free-box ?box))
)
)

```

## Scarico del contenuto da una scatola

Lo scarico del contenuto della scatola deve essere fatto solo in presenza di una persona che necessita di quel contenuto e per tale motivo questa azione può essere effettuata solo se la persona, il carrier, la scatola e il robot sono nella stessa zona, inoltre è necessario anche che la persona abbia bisogno del contenuto in questione. Gli effetti di questa azione saranno che la scatola sarà vuota e che la persona non avrà più necessità del contenuto.

```

(:action drop-content-of-box
  :parameters (?box - box ?cont - content ?rob - robot ?per - person ?zone -
zone
)
  :precondition (and
    (box-filled ?box ?cont)
    (at ?box ?zone)
    (at ?rob ?zone)
    (at ?per ?zone)
    (person-needs-content ?per ?cont)
  )
  :effect (and
    (not (box-filled ?box ?cont))
    (free-box ?box)
    (person-has-content ?per ?cont)
    (not (person-needs-content ?per ?cont))
  )
)
)

```

## Carico di una scatola sul carrello

Il carico di una scatola sul carrello può essere effettuato solo se il carrello presenta una slot libero e insieme alla scatola e al robot si trovano nella stessa zona. Inoltre, per come è stato definito il dominio, ovvero avendo i compartimenti (slot) dichiarati in modo assestante è anche necessario che il compartimento su cui si carica la scatola faccia parte del carrello specificato. Gli

effetti di tale operazione saranno che la scatola non si troverà più sulla zona indicata ma sul carrello (carrier) e il compartimento non sarà più libero.

```
(:action put-box-on-carrier
  :parameters (?box - box ?car - carrier ?rob - robot ?zone - zone ?slot -
slot
)
  :precondition (and
    (at ?box ?zone)
    (at ?car ?zone)
    (at ?rob ?zone)
    (slot-of-carrier ?slot ?car)
    (free-slot ?slot)
  )
  :effect (and
    (not (at ?box ?zone))
    (box-on-carrier ?box ?car)
    (not (free-slot ?slot))
  )
)
```

## Scarico di una scatola dal carrello

Lo scarico di una scatola dal carrello è effettuato in maniera esattamente opposta al carico, ovvero è necessario che il robot e il carrello siano nella stessa zona, la scatola sia sul carrello e il compartimento sia occupato dalla scatola. Gli effetti di questa operazione saranno che la scatola si troverà sulla zona specificata e il compartimento sarà libero.

```
(:action drop-box-of-carrier
  :parameters (?box - box ?car - carrier ?rob - robot ?zone - zone ?slot -
slot
)
  :precondition (and
    (at ?car ?zone)
    (at ?rob ?zone)
    (box-on-carrier ?box ?car)
    (slot-of-carrier ?slot ?car)
    (not (free-slot ?slot))
  )
  :effect (and
    (at ?box ?zone)
    (not (box-on-carrier ?box ?car))
    (free-slot ?slot)
  )
)
```



## Movimento del robot

Il movimento del robot può avvenire in qualsiasi situazione a patto che esso si trovi in una determinata zona, l'effetto di questa operazione è che il robot non si troverà più nella zona precedente ma in quella successiva.

```
(:action move-robot
  :parameters (?rob - robot ?from - zone ?to - zone)
  :precondition (and (at ?rob ?from))
  :effect (and
    (not (at ?rob ?from))
    (at ?rob ?to)
  )
)
```

## Movimento del robot con carrello

In maniera del tutto analoga il movimento del robot con il carrello è possibile solo se anche il carrello si trova nella stessa posizione del robot e l'effetto sarà che entrambi verranno spostati nella zona successiva.

```
(:action move-robot-with-carrier
  :parameters (?rob - robot ?car - carrier ?from - zone ?to - zone)
  :precondition (and
    (at ?rob ?from)
    (at ?car ?from)
  )
  :effect (and
    (not (at ?rob ?from))
    (at ?rob ?to)
    (not (at ?car ?from))
    (at ?car ?to)
  )
)
```

## Note sul dominio

È bene specificare che per ragioni di efficienza e semplicità si è deciso di non formalizzare un'associazione tra compartimento e scatola ma tenere traccia solo dello stato dei compartimenti del carrello, ovvero quanti di questi sono pieni e quanti sono liberi. Infatti, l'idea alla base era quella di definire semplicemente la

capienza del carrello senza specificare in che ordine questo venisse riempito o svuotato.

Per i dettagli implementativi del dominio descritto rimandiamo comunque alla visione del file *rescue\_domain.pddl* presente in allegato.

# Classical planning

Al fine di realizzare un planner che risolva le istanze dei problemi nelle sezioni successive si è utilizzata la libreria [PDDL4J](#) la quale consente di avere dei planner già pronti e da configurare o di costruirne uno personalizzato mediante l'estensione di alcune classi in linguaggio Java.

## Planner

Il planner definito implementa l'algoritmo A\* il quale combina insieme le strategie Greedy e UCS definendo un ordinamento per la ricerca pari alla somma della funzione euristica ( $h$ ) e della funzione di costo ( $g$ ). Nello specifico la versione implementata ha alcuni cambiamenti e peculiarità rispetto all'algoritmo A\* classico riassunti nei seguenti punti:

1. La funzione usata per ordinare la pila di nodi da esplorare differisce dalla classica funzione ( $f$ ) usata in A\* dato che l'euristica ha un fattore moltiplicativo ( $w$ ) che ne definisce il peso:  $f(x) = w \cdot h(x) + g(x)$ . Tale fattore, in tutte le esecuzioni successive, verrà posto uguale a 25.
2. È stato impostato un limite alla profondità della ricerca per evitare possibili stati di errore dovuti all'esaurimento della memoria centrale della macchina a disposizione. Tale limite, in tutte le esecuzioni successive, rimarrà fissato a 200 ma non verrà mai sfiorato dato che le soluzioni trovate saranno sempre a livelli superiori.
3. La funzione di costo ( $g$ ) è data dal numero di nodi esplorati fino al momento, ovvero dalla profondità.

L'implementazione di tale planner è stata resa possibile estendendo la classe *AbstractPlanner* della libreria PDDL4J con la classe *AstarCustom*. Per ragioni di brevità si riportano di seguito solo i metodi *solve(Problem)* e *astar(Problem)* i quali implementano l'algoritmo di ricerca descritto sopra e rimanda ad una visione del file *AstarCustom.java* presente in allegato.

```
public Plan solve(final Problem problem) {
    LOGGER.info("* Starting A* search \n");
    final long begin = System.currentTimeMillis();
    // Start search with custom A* algorithm
```

```

        final Plan plan = this.astar(problem);
        final long end = System.currentTimeMillis();
        if (plan != null) {
            // Search successes set statistics
            LOGGER.info("* A* search succeeded \n");
            this.getStatistics().setTimeToSearch(end - begin);
            this.getStatistics().setMemoryUsedToSearch(Math.abs(endMemory -
beginMemory));
        }
        else {
            LOGGER.info("* A* search failed \n");
        }
        // If fail return null plan
        return plan;
    }
}

```

```

public Plan astar(Problem problem) {
    // Check if well formed
    if (!this.isSupported(problem)) {
        return null;
    }
    Runtime runtime = Runtime.getRuntime();
    this.beginMemory = runtime.totalMemory() - runtime.freeMemory();
    final StateHeuristic heuristic = this.getHeuristics(problem);
    // Build the initial state of the problem
    final State init = new State(problem.getInitialState());
    // Define a set of already explored node
    final Set<Node> exploredNode = new HashSet<>();
    final double weight = this.getHeuristicWeight();
    // Define a fringe as priority queue based of weighted-f function
    final PriorityQueue<Node> fringe = new PriorityQueue<>(100, new
Comparator<Node>() {
        @Override
        public int compare(Node n1, Node n2) {
            double f1 = weight * n1.getHeuristic() + n1.getCost();
            double f2 = weight * n2.getHeuristic() + n2.getCost();
            return Double.compare(f1, f2);
        }
    });
    // Build the root of the searching three
    final Node root = new Node(init, null, -1, 0, heuristic.estimate(init,
problem.getGoal()));
    fringe.add(root);
    Plan plan = null;
    // Start the search
    while (!fringe.isEmpty() && plan == null) {
        // Remove first node from the fringe
        final Node current = fringe.poll();
        // Ignore the node if depth exceeded
        if (current.getDepth() > this.maxDepth) {
            continue;
        }
        exploredNode.add(current);
        if (current.satisfy(problem.getGoal())) {
            // Problem solved
            LOGGER.info("Found goal at depth " + current.getDepth() +
"\n");
        }
    }
}

```

```

        this.endMemory = runtime.totalMemory() -
runtime.freeMemory();
        return this.extractPlan(current, problem);
    }
    else {
        for (int i = 0; i < problem.getActions().size(); i++) {
            // We get the actions of the problem
            Action a = problem.getActions().get(i);
            if (a.isApplicable(current)) {
                // If action a is applicable to current node
generate child node
                Node next = new Node(current);
                next.setDepth(current.getDepth() + 1);
                // Get requirements and effects of the action a
applied
                final List<ConditionalEffect> effects =
a.getConditionalEffects();
                for (ConditionalEffect ce : effects) {
                    // In the current state satisfy all the
preconditions apply the effects on the child node
                    if (current.satisfy(ce.getCondition()))
{
                        next.apply(ce.getEffect());
                    }
                }
                // if the new node (state) is not explored yet
add the child into the fringe
                if (!exploredNode.contains(next)) {
                    // Define cost as the number of the
nodes already explored
                    next.setCost(current.getCost() + 1);
                    next.setParent(current);
                    // Set the applied action (encoded as
integer) that generate child node
                    next.setAction(i);
                    // Set the estimate distance to the goal
according to the current heuristics
                    next.setHeuristic(heuristic.estimate(next, problem.getGoal()));
                    // Add child node to the fringe to be
explored soon
                    fringe.add(next);
                }
            }
        }
    }
    this.endMemory = runtime.totalMemory() - runtime.freeMemory();
    return plan;
}

```

Inoltre, si è reso necessario definire anche la classe Node, la quale estende la classe State di PDDL4J e rappresenta un nodo dell'albero di ricerca generato. Tale classe contiene le informazioni necessarie alla ricerca come il costo del nodo raggiunto, la profondità, l'azione che l'ha generato, etc.

Tuttavia, data la sua trivialità si riporta di seguito solo la parte iniziale di tale classe rimandando alla visione del file *Node.java* presente in allegato.

```
public final class Node extends State {  
  
    private Node parent;  
    private int action;  
    private double cost;  
    private double heuristic;  
    private int depth;  
    ...  
  
}
```

## Euristica

L'euristica realizzata e usata per risolvere le istanze dei problemi descritti nella traccia presenti nelle sezioni successive è una versione modificata dell'euristica *Adjusted Sum* presente nella libreria PDDL4J la quale tiene in considerazione sia gli effetti positivi che negativi dei goal, inoltre implementa una rappresentazione a bit-vector<sup>1</sup> per i fluents<sup>2</sup>. Al contrario, l'euristica realizzata considera solo gli effetti positivi prendendo quindi in considerazione soltanto una versione rilassata del problema.

Nello specifico la funzione di stima della distanza dal goal (*estimate(Node)*) calcola la funzione  $h(x) = \text{sumValue}(x) + (\text{level} - \text{maxValue}())$  se il goal è raggiungibile dallo stato attuale altrimenti ritorna infinito. In particolare il significato dei componenti della funzione *h* è il seguente:

1. **Sum value:** è la somma dei conteggi di quante volte appaiono i fluent positivi nel grafo di planning rilassato (costruito dal nodo attuale) che compaiono anche nel goal. Tale valore fornisce una metrica approssimata alla distanza dal goal sebbene un fluent positivo possa apparire più volte nel conteggio. Infatti, tale valore è usato proprio a questo scopo dell'euristica di tipo *Sum*.
2. **Max value:** è il massimo numero di volte in cui un fluent positivo che appare nel grafo di planning rilassato (costruito dal nodo attuale) compare

---

<sup>1</sup> Un array di bit usato per rappresentare booleani.

<sup>2</sup> Un fluent è una proposizione logica il cui valore è soggetto a cambiamento durante la fase di planning.

anche nel goal.

3. **Level:** è il livello di connettività<sup>3</sup> del grafo di planning rilassato costruito a partire dal nodo attuale (quello del quale stiamo calcolando l'euristica). Ovvero ogni nodo del grafo ha un livello di connettività almeno pari a *level*.

Per comprendere meglio il funzionamento di tale meccanismo si riporta di seguito il codice della classe *PositiveAdjustedSum* opportunamente commentato il cui codice sorgente è apposto in allegato e al quale rimandiamo la visione per maggior chiarezza.

```
public class PositiveAdjustedSum extends RelaxedGraphHeuristic {

    protected PositiveAdjustedSum(Problem problem) {
        super(problem);
        super.setAdmissible(false);
    }

    public int estimate(State state, Condition goal) {
        super.setGoal(goal);
        /* A relaxed planning graph is a graph that is used to represent
the state space of a planning problem.
        * It is a relaxation of the standard planning graph, in that it
does not consider all possible ways
        * in which the state of the world can change. This makes it more
efficient to construct and search,
        * but it also means that it may not be able to find the optimal
solution to the problem.
        *
        * The relaxed problem is obtained by removing the delete effects
from all actions.
        *
        * Get the connectivity level of the relaxed planning graph i.e.
every edge connectivity is at least equal to such level */
        final int level = super.expandRelaxedPlanningGraph(state);
        return super.isGoalReachable()
            ? this.getSumValue() + (level - this.getMaxValue()) :
Integer.MAX_VALUE;
    }

    public double estimate(Node node, Condition goal) {
        return estimate((State) node, goal);
    }

    /* Compute the sum of the times each proposition appear true in the
relaxed planning
        * graph builded. That define a metric over the distance to the goal.
        */
}
```

---

<sup>3</sup> Il livello di connettività di un grafo è il numero minimo di vertici che devono essere rimossi per disconnettere il grafo. Un grafo si dice k-connesso se rimane connesso dopo la rimozione di k-1 vertici. Il massimo di tale numero k è chiamato connettività del grafico.

```

    * In fact, this is the values used in the sum heuristics */
    protected final int getSumValue() {
        int value = 0;
        // Get goal's positive fluent
        // A fluent is a proposition whose truth value changes during
        planning process
        final BitVector pGoal = super.getGoal().getPositiveFluents();
        for (int g = pGoal.nextSetBit(0); g >= 0; g = pGoal.nextSetBit(g +
1)) {
            /* Get the level of appearance (i.e. the times) of the
            proposition identified by a
            * fluent g appear true in the relaxed panning graph builded
            */
            value += this.pPropLevel[g];
        }
        return value;
    }

    protected int getMaxValue() {
        int max = Integer.MIN_VALUE;
        final BitVector pGoal = super.getGoal().getPositiveFluents();
        for (int g = pGoal.nextSetBit(0); g >= 0; g = pGoal.nextSetBit(g +
1)) {
            final int gl = this.pPropLevel[g];
            if (gl > max) {
                max = gl;
            }
        }
        // Return the max times of a proposition appear true in the the
        relaxed panning graph builded
        return max;
    }
}

```

## Instance 1

Le condizioni iniziali della prima istanza riportata nella traccia sono:

1. Cinque scatole posizionate nella zona *depot*.
2. Cibo, attrezzi e medicine in numero sufficiente posizionati al *depot*.
3. Le persone *p1* e *p2* nella stessa zona, diciamo *zone1*, e *p3* in una zona diversa, diciamo *zone2*. *p1* richiede di cibo e medicine, *p2* richiede solo medicine e *p3* solo cibo.
4. Un singolo robot e un singolo carrello al *depot*.
5. Il carrello ha capacità di carico di massimo quattro scatole.

Per modellare tale problema si è definito un file di problema in linguaggio PDDL nel quale mediante la dichiarazione `objects` si istanziano gli oggetti



necessari al problema in accordo con la gerarchia definita. Successivamente mediante la dichiarazione `init` e `goal` si specificano le condizioni iniziali, predicati veri all'inizio, e gli obiettivi, predicati che devono essere veri alla fine. Di seguito è riportato il codice del file *rescue\_instance1.pddl* il quale è presente in allegato.

```
(define (problem rescue_inst_1)
  (:domain rescue_sys)

  (:objects
    depot zone1 zone2 - zone
    b1 b2 b3 b4 b5 - box
    p1 p2 p3 - person
    r - robot
    car - carrier
    sl1 sl2 sl3 sl4 - slot
    drugs food tools - content
  )

  (:init
    (at b1 depot)
    (at b2 depot)
    (at b3 depot)
    (at b4 depot)
    (at b5 depot)
    (at r depot)
    (at car depot)
    (at food depot)
    (at drugs depot)
    (at tools depot)
    (at p1 zone1)
    (at p2 zone1)
    (at p3 zone2)
    (person-needs-content p1 food)
    (person-needs-content p1 drugs)
    (person-needs-content p2 drugs)
    (person-needs-content p3 food)
    (free-box b1)
    (free-box b2)
    (free-box b3)
    (free-box b4)
    (free-box b5)
    (free-slot sl1)
    (free-slot sl2)
    (free-slot sl3)
    (free-slot sl4)
    (slot-of-carrier sl1 car)
    (slot-of-carrier sl2 car)
    (slot-of-carrier sl3 car)
    (slot-of-carrier sl4 car)
  )

  (:goal
    (and
      ; Person 1
```

```

        (person-has-content p1 food)
        (person-has-content p1 drugs)
        ; Person 2
        (person-has-content p2 drugs)
        ; Person 3
        (person-has-content p3 food)
    )
)
)

```

La risoluzione di questa prima istanza con il planner e l'euristica definiti in precedenza avviene in tempi brevissimi (*1.9s*) e con un uso di memoria centrale decisamente minimo di soli *56.66 MBytes*. L'output fornito è il seguente

```
problem instantiation done successfully (173 actions, 68 fluents)
```

```

* Starting A* search
Using custom heuristics
Found goal at depth 24
* A* search succeeded

```

```
found plan as follows:
```

```

00: (      put-content-in-box b1 drugs r depot) [0]
01: (      put-box-on-carrier b1 car r depot p13) [0]
02: (move-robot-with-carrier r car depot zone1) [0]
03: (      drop-box-of-carrier b1 car r zone1 p13) [0]
04: (move-robot-with-carrier r car zone1 depot) [0]
05: (      put-content-in-box b2 food r depot) [0]
06: (      put-box-on-carrier b2 car r depot p11) [0]
07: (move-robot-with-carrier r car depot zone1) [0]
08: (      drop-box-of-carrier b2 car r zone1 p11) [0]
09: (      drop-content-of-box b2 food r p1 zone1) [0]
10: (move-robot-with-carrier r car zone1 depot) [0]
11: (      put-content-in-box b3 food r depot) [0]
12: (      put-box-on-carrier b3 car r depot p11) [0]
13: (move-robot-with-carrier r car depot zone2) [0]
14: (      drop-box-of-carrier b3 car r zone2 p11) [0]
15: (      drop-content-of-box b3 food r p3 zone2) [0]
16: (move-robot-with-carrier r car zone2 zone1) [0]
17: (      drop-content-of-box b1 drugs r p1 zone1) [0]
18: (move-robot-with-carrier r car zone1 depot) [0]
19: (      put-content-in-box b4 drugs r depot) [0]
20: (      put-box-on-carrier b4 car r depot p11) [0]
21: (move-robot-with-carrier r car depot zone1) [0]
22: (      drop-box-of-carrier b4 car r zone1 p11) [0]
23: (      drop-content-of-box b4 drugs r p2 zone1) [0]

```

```

time spent:      0.05 seconds parsing
                 0.10 seconds encoding
                 1.75 seconds searching
                 1.90 seconds total time

```

```

memory used:      0.92 MBytes for problem representation
                  55.74 MBytes for searching
                  56.66 MBytes total

```

## Instance 2

Le condizioni iniziali della seconda istanza riportata nella traccia sono:

1. Due robot e due carrelli posizionati al depot.
2. Tre scatole posizionate al depot.
3. Capacità del carrello pari a due.
4. Sei persone:  $p1$  richiede cibo o attrezzi,  $p2$  richiede medicine,  $p3$  richiede medicine,  $p4$  richiede medicine e cibo,  $p5$  e  $p6$  richiedono tutto.
5. Le prime due persone sono nella stessa zona mentre le altre in posizione diversa.

Per modellare il problema si è definito un file di problema *rescue\_instance2.pddl* del tutto analogo al precedente ma con specifiche diverse. Tuttavia, si sottolinea che la realizzazione della condizione esclusiva sulle richieste della persona  $p1$  non è stato realizzabile dal momento che il parser intrinseco nella libreria *PDDL4J* non permette di inserire una condizione in *or* nei goal. Per tale motivo le condizioni sono state messe in *and* al resto dei goal.

Di seguito si riporta tale file presente sempre in allegato al quale rimandiamo la visione.

```
(define (problem rescue_inst_2)
  (:domain rescue_sys)
  (:requirements :disjunctive-preconditions)

  (:objects
    depot zone1 zone2 zone3 zone4 zone5 - zone
    b1 b2 b3 - box
    p1 p2 p3 p4 p5 p6 - person
    r1 r2 - robot
    car1 car2 - carrier
    sl1car1 sl2car1 sl1car2 sl2car2 - slot
    drugs food tools - content
  )

  (:init
    (at b1 depot)
    (at b2 depot)
    (at b3 depot)
    (at r1 depot)
    (at r2 depot)
```

```

(at car1 depot)
(at car2 depot)
(at food depot)
(at drugs depot)
(at tools depot)
(at p1 zone1)
(at p2 zone1)
(at p3 zone2)
(at p4 zone3)
(at p5 zone4)
(at p6 zone5)
(person-needs-content p1 food)
(person-needs-content p1 tools)
(person-needs-content p2 drugs)
(person-needs-content p3 drugs)
(person-needs-content p4 food)
(person-needs-content p4 drugs)
(person-needs-content p5 food)
(person-needs-content p5 drugs)
(person-needs-content p5 tools)
(person-needs-content p6 food)
(person-needs-content p6 drugs)
(person-needs-content p6 tools)
(free-box b1)
(free-box b2)
(free-box b3)
(free-slot sl1car1)
(free-slot sl2car1)
(free-slot sl1car2)
(free-slot sl2car2)
(slot-of-carrier sl1car1 car1)
(slot-of-carrier sl2car1 car1)
(slot-of-carrier sl1car2 car2)
(slot-of-carrier sl2car2 car2)
)

(:goal
  (and
    ; Person 1
    (person-has-content p1 food)
    (person-has-content p1 tools)
    ; Person 2
    (person-has-content p2 drugs)
    ; Person 3
    (person-has-content p3 drugs)
    ; Person 4
    (person-has-content p4 food)
    (person-has-content p4 drugs)
    ; Person 5
    (person-has-content p5 food)
    (person-has-content p5 drugs)
  )
)

```

```

        (person-has-content p5 tools)
        ; Person 6
        (person-has-content p6 food)
        (person-has-content p6 drugs)
        (person-has-content p6 tools)
    )
)
)

```

A differenza del caso precedente la risoluzione di questa seconda istanza richiede qualche risorsa in più dal momento che termina in circa 30 secondi e occupa circa 250 MBytes. Il risultato del planner è il seguente.

```

problem instantiation done successfully (594 actions, 101 fluents)

* Starting A* search
Using custom heuristics
Found goal at depth 88
* A* search succeeded

found plan as follows:

00: ( put-content-in-box b3 tools r2 depot) [0]
01: ( put-box-on-carrier b3 car1 r1 depot s11car1) [0]
02: ( move-robot-with-carrier r1 car1 depot zone1) [0]
03: (drop-box-of-carrier b3 car1 r1 zone1 s11car1) [0]
04: ( drop-content-of-box b3 tools r1 p1 zone1) [0]
05: ( put-content-in-box b1 tools r2 depot) [0]
06: ( move-robot-with-carrier r1 car1 zone1 depot) [0]
07: ( put-box-on-carrier b1 car1 r1 depot s11car1) [0]
08: ( move-robot-with-carrier r1 car1 depot zone4) [0]
09: (drop-box-of-carrier b1 car1 r1 zone4 s11car1) [0]
10: ( drop-content-of-box b1 tools r1 p5 zone4) [0]
11: ( put-box-on-carrier b1 car1 r1 zone4 s11car1) [0]
12: ( move-robot-with-carrier r1 car1 zone4 depot) [0]
13: ( put-content-in-box b2 drugs r1 depot) [0]
14: ( put-box-on-carrier b2 car2 r1 depot s11car2) [0]
15: ( move-robot-with-carrier r1 car2 depot zone1) [0]
16: (drop-box-of-carrier b2 car2 r1 zone1 s11car2) [0]
17: ( drop-content-of-box b2 drugs r1 p2 zone1) [0]
18: ( put-box-on-carrier b2 car2 r1 zone1 s11car2) [0]
19: ( move-robot-with-carrier r1 car2 zone1 depot) [0]
20: (drop-box-of-carrier b1 car1 r1 depot s11car1) [0]
21: ( put-content-in-box b1 drugs r1 depot) [0]
22: ( put-box-on-carrier b1 car1 r1 depot s11car1) [0]
23: ( move-robot-with-carrier r1 car1 depot zone2) [0]
24: (drop-box-of-carrier b1 car1 r1 zone2 s11car1) [0]
25: ( drop-content-of-box b1 drugs r1 p3 zone2) [0]
26: ( put-box-on-carrier b1 car1 r1 zone2 s11car1) [0]

```

```

27: (drop-box-of-carrier b2 car2 r2 depot s11car2) [0]
28: ( move-robot-with-carrier r1 car1 zone2 depot) [0]
29: (      put-content-in-box b2 drugs r1 depot) [0]
30: ( put-box-on-carrier b2 car2 r1 depot s11car2) [0]
31: ( move-robot-with-carrier r1 car2 depot zone3) [0]
32: (drop-box-of-carrier b2 car2 r1 zone3 s11car2) [0]
33: (      drop-content-of-box b2 drugs r1 p4 zone3) [0]
34: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
35: ( put-box-on-carrier b2 car2 r1 zone3 s11car2) [0]
36: ( move-robot-with-carrier r1 car2 zone3 depot) [0]
37: (      put-content-in-box b1 drugs r1 depot) [0]
38: ( put-box-on-carrier b1 car1 r1 depot s11car1) [0]
39: ( move-robot-with-carrier r1 car1 depot zone4) [0]
40: (drop-box-of-carrier b1 car1 r1 zone4 s11car1) [0]
41: (      drop-content-of-box b1 drugs r1 p5 zone4) [0]
42: ( put-box-on-carrier b1 car1 r1 zone4 s11car1) [0]
43: (drop-box-of-carrier b2 car2 r2 depot s11car2) [0]
44: ( move-robot-with-carrier r1 car1 zone4 depot) [0]
45: (      put-content-in-box b2 drugs r1 depot) [0]
46: ( put-box-on-carrier b2 car2 r1 depot s11car2) [0]
47: ( move-robot-with-carrier r1 car2 depot zone5) [0]
48: (drop-box-of-carrier b2 car2 r1 zone5 s11car2) [0]
49: (      drop-content-of-box b2 drugs r1 p6 zone5) [0]
50: ( put-box-on-carrier b2 car2 r1 zone5 s11car2) [0]
51: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
52: ( move-robot-with-carrier r1 car2 zone5 depot) [0]
53: (      put-content-in-box b1 food r1 depot) [0]
54: ( put-box-on-carrier b1 car1 r1 depot s11car1) [0]
55: ( move-robot-with-carrier r1 car1 depot zone1) [0]
56: (drop-box-of-carrier b1 car1 r1 zone1 s11car1) [0]
57: (      drop-content-of-box b1 food r1 p1 zone1) [0]
58: ( put-box-on-carrier b1 car1 r1 zone1 s11car1) [0]
59: (drop-box-of-carrier b2 car2 r2 depot s11car2) [0]
60: ( move-robot-with-carrier r1 car1 zone1 depot) [0]
61: (      put-content-in-box b2 food r1 depot) [0]
62: ( put-box-on-carrier b2 car2 r1 depot s11car2) [0]
63: ( move-robot-with-carrier r1 car2 depot zone3) [0]
64: (drop-box-of-carrier b2 car2 r1 zone3 s11car2) [0]
65: (      drop-content-of-box b2 food r1 p4 zone3) [0]
66: ( put-box-on-carrier b2 car2 r1 zone3 s11car2) [0]
67: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
68: ( move-robot-with-carrier r1 car2 zone3 depot) [0]
69: (      put-content-in-box b1 food r1 depot) [0]
70: ( put-box-on-carrier b1 car1 r1 depot s11car1) [0]
71: ( move-robot-with-carrier r1 car1 depot zone4) [0]
72: (drop-box-of-carrier b1 car1 r1 zone4 s11car1) [0]
73: (      drop-content-of-box b1 food r1 p5 zone4) [0]
74: ( put-box-on-carrier b1 car1 r1 zone4 s11car1) [0]
75: (drop-box-of-carrier b2 car2 r2 depot s11car2) [0]
76: ( move-robot-with-carrier r1 car1 zone4 depot) [0]
77: (      put-content-in-box b2 food r1 depot) [0]

```

```

78: ( put-box-on-carrier b2 car2 r1 depot s11car2) [0]
79: ( move-robot-with-carrier r1 car2 depot zone5) [0]
80: (drop-box-of-carrier b2 car2 r1 zone5 s11car2) [0]
81: (      drop-content-of-box b2 food r1 p6 zone5) [0]
82: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
83: (      put-content-in-box b1 tools r2 depot) [0]
84: ( put-box-on-carrier b1 car1 r2 depot s11car1) [0]
85: ( move-robot-with-carrier r2 car1 depot zone5) [0]
86: (drop-box-of-carrier b1 car1 r1 zone5 s11car1) [0]
87: (      drop-content-of-box b1 tools r1 p6 zone5) [0]

```

```

time spent:      0.05 seconds parsing
                 0.17 seconds encoding
                 29.74 seconds searching
                 29.95 seconds total time

```

```

memory used:     2.74 MBytes for problem representation
                 249.83 MBytes for searching
                 252.58 MBytes total

```

## Instance 3 (Bonus)

Le condizioni iniziali della terza istanza riportata nella traccia sono:

1. Quattro scatole al depot
2. Otto persone: p1 richiede cibo o attrezzi, p2 richiede medicine, p3 richiede medicine, p4 richiede medicine e cibo, p5, p6, p7 e p8 richiedono tutto.
3. Le prime due persone sono posizionate nella stessa zona le restanti in zone diverse
4. Due robot
5. Due carrelli con capienza due

Per modellare il problema si è definito un file di problema *rescue\_instance3.pddl* del tutto analogo al precedente ma con specifiche diverse reperibile sempre in allegato. Come nel caso precedente anche in questa terza istanza è stato impossibile inserire una condizione in *or* nei goal per le motivazioni discusse al punto precedente e le tali condizioni sono state poste in *and* con le restanti.

```

(define (problem rescue_inst_3)
  (:domain rescue_sys)

```

```

(:requirements :disjunctive-preconditions)

(:objects
  depot zone1 zone2 zone3 zone4 zone5 zone6 zone7 - zone
  b1 b2 b3 b4 - box
  p1 p2 p3 p4 p5 p6 p7 p8 - person
  r1 r2 - robot
  car1 car2 - carrier
  s11car1 s12car1 s11car2 s12car2 - slot
  drugs food tools - content
)

(:init
  (at b1 depot)
  (at b2 depot)
  (at b3 depot)
  (at b4 depot)
  (at r1 depot)
  (at r2 depot)
  (at car1 depot)
  (at car2 depot)
  (at food depot)
  (at drugs depot)
  (at tools depot)
  (at p1 zone1)
  (at p2 zone1)
  (at p3 zone2)
  (at p4 zone3)
  (at p5 zone4)
  (at p6 zone5)
  (at p7 zone6)
  (at p8 zone7)
  (person-needs-content p1 food)
  (person-needs-content p1 tools)
  (person-needs-content p2 drugs)
  (person-needs-content p3 drugs)
  (person-needs-content p4 food)
  (person-needs-content p4 drugs)
  (person-needs-content p5 food)
  (person-needs-content p5 drugs)
  (person-needs-content p5 tools)
  (person-needs-content p6 food)
  (person-needs-content p6 drugs)
  (person-needs-content p6 tools)
  (person-needs-content p7 food)
  (person-needs-content p7 drugs)
  (person-needs-content p7 tools)
  (person-needs-content p8 food)
  (person-needs-content p8 drugs)
  (person-needs-content p8 tools)
  (free-box b1)
  (free-box b2)
  (free-box b3)
  (free-box b4)
  (free-slot s11car1)
  (free-slot s12car1)
  (free-slot s11car2)
  (free-slot s12car2)

```



```

        (slot-of-carrier sl1car1 car1)
        (slot-of-carrier sl2car1 car1)
        (slot-of-carrier sl1car2 car2)
        (slot-of-carrier sl2car2 car2)
    )

    (:goal
      (and
        ; Person 1
        (person-has-content p1 food)
        (person-has-content p1 tools)
        ; Person 2
        (person-has-content p2 drugs)
        ; Person 3
        (person-has-content p3 drugs)
        ; Person 4
        (person-has-content p4 food)
        (person-has-content p4 drugs)
        ; Person 5
        (person-has-content p5 food)
        (person-has-content p5 drugs)
        (person-has-content p5 tools)
        ; Person 6
        (person-has-content p6 food)
        (person-has-content p6 drugs)
        (person-has-content p6 tools)
        ; Person 7
        (person-has-content p7 food)
        (person-has-content p7 drugs)
        (person-has-content p7 tools)
        ; Person 8
        (person-has-content p8 food)
        (person-has-content p8 drugs)
        (person-has-content p8 tools)
      )
    )
  )
)

```

La risoluzione di questa terza istanza richiede un numero sostanzialmente più elevato di risorse di calcolo impiegando quasi 3 minuti e mezzo e usando circa un GByte di memoria centrale. L'output di tale risoluzione con planner ed euristica definiti è riportato di seguito.

```

problem instantiation done successfully (1064 actions, 143 fluents)

* Starting A* search
Using custom heuristics
Found goal at depth 134
* A* search succeeded

found plan as follows:

000: (      put-content-in-box b1 drugs r2 depot) [0]
001: ( put-box-on-carrier b1 car1 r2 depot sl2car1) [0]

```

```

002: ( move-robot-with-carrier r1 car1 depot zone1) [0]
003: (drop-box-of-carrier b1 car1 r1 zone1 s12car1) [0]
004: (      drop-content-of-box b1 drugs r1 p2 zone1) [0]
005: (          put-content-in-box b3 tools r2 depot) [0]
006: ( put-box-on-carrier b3 car2 r2 depot s11car2) [0]
007: ( move-robot-with-carrier r1 car1 zone1 depot) [0]
008: ( move-robot-with-carrier r1 car2 depot zone1) [0]
009: (drop-box-of-carrier b3 car2 r1 zone1 s11car2) [0]
010: (      drop-content-of-box b3 tools r1 p1 zone1) [0]
011: (          put-content-in-box b2 drugs r2 depot) [0]
012: ( put-box-on-carrier b2 car1 r2 depot s11car1) [0]
013: ( move-robot-with-carrier r1 car2 zone1 depot) [0]
014: ( move-robot-with-carrier r1 car1 depot zone2) [0]
015: (drop-box-of-carrier b2 car1 r1 zone2 s11car1) [0]
016: (      drop-content-of-box b2 drugs r1 p3 zone2) [0]
017: ( put-box-on-carrier b2 car1 r1 zone2 s11car1) [0]
018: ( move-robot-with-carrier r1 car1 zone2 depot) [0]
019: (          put-content-in-box b4 drugs r1 depot) [0]
020: ( put-box-on-carrier b4 car2 r1 depot s11car2) [0]
021: ( move-robot-with-carrier r1 car2 depot zone3) [0]
022: (drop-box-of-carrier b4 car2 r1 zone3 s11car2) [0]
023: (      drop-content-of-box b4 drugs r1 p4 zone3) [0]
024: ( put-box-on-carrier b4 car2 r1 zone3 s12car2) [0]
025: (drop-box-of-carrier b2 car1 r2 depot s11car1) [0]
026: ( move-robot-with-carrier r1 car2 zone3 depot) [0]
027: (          put-content-in-box b2 drugs r1 depot) [0]
028: ( put-box-on-carrier b2 car1 r1 depot s11car1) [0]
029: ( move-robot-with-carrier r1 car1 depot zone4) [0]
030: (drop-box-of-carrier b2 car1 r1 zone4 s11car1) [0]
031: (      drop-content-of-box b2 drugs r1 p5 zone4) [0]
032: ( put-box-on-carrier b2 car1 r1 zone4 s11car1) [0]
033: (drop-box-of-carrier b4 car2 r2 depot s12car2) [0]
034: ( move-robot-with-carrier r1 car1 zone4 depot) [0]
035: (          put-content-in-box b4 drugs r1 depot) [0]
036: ( put-box-on-carrier b4 car2 r1 depot s11car2) [0]
037: ( move-robot-with-carrier r1 car2 depot zone5) [0]
038: (drop-box-of-carrier b4 car2 r1 zone5 s11car2) [0]
039: (      drop-content-of-box b4 drugs r1 p6 zone5) [0]
040: ( put-box-on-carrier b4 car2 r1 zone5 s11car2) [0]
041: (drop-box-of-carrier b2 car1 r2 depot s11car1) [0]
042: ( move-robot-with-carrier r1 car2 zone5 depot) [0]
043: (          put-content-in-box b2 drugs r1 depot) [0]
044: ( put-box-on-carrier b2 car1 r1 depot s11car1) [0]
045: ( move-robot-with-carrier r1 car1 depot zone6) [0]
046: (drop-box-of-carrier b2 car1 r1 zone6 s11car1) [0]
047: (      drop-content-of-box b2 drugs r1 p7 zone6) [0]
048: (drop-box-of-carrier b4 car2 r2 depot s11car2) [0]
049: ( put-box-on-carrier b2 car1 r1 zone6 s11car1) [0]
050: ( move-robot-with-carrier r1 car1 zone6 depot) [0]
051: (          put-content-in-box b4 drugs r1 depot) [0]
052: ( put-box-on-carrier b4 car2 r1 depot s11car2) [0]

```

```

053: ( move-robot-with-carrier r1 car2 depot zone7) [0]
054: (drop-box-of-carrier b4 car2 r1 zone7 s11car2) [0]
055: (      drop-content-of-box b4 drugs r1 p8 zone7) [0]
056: ( put-box-on-carrier b4 car2 r1 zone7 s11car2) [0]
057: (drop-box-of-carrier b2 car1 r2 depot s11car1) [0]
058: ( move-robot-with-carrier r1 car2 zone7 depot) [0]
059: (      put-content-in-box b2 food r1 depot) [0]
060: ( put-box-on-carrier b2 car1 r1 depot s11car1) [0]
061: ( move-robot-with-carrier r1 car1 depot zone1) [0]
062: (drop-box-of-carrier b2 car1 r1 zone1 s11car1) [0]
063: (      drop-content-of-box b2 food r1 p1 zone1) [0]
064: ( put-box-on-carrier b1 car1 r1 zone1 s11car1) [0]
065: (drop-box-of-carrier b4 car2 r2 depot s11car2) [0]
066: ( move-robot-with-carrier r1 car1 zone1 depot) [0]
067: (      put-content-in-box b4 food r1 depot) [0]
068: ( put-box-on-carrier b4 car2 r1 depot s11car2) [0]
069: ( move-robot-with-carrier r1 car2 depot zone3) [0]
070: (drop-box-of-carrier b4 car2 r1 zone3 s11car2) [0]
071: (      drop-content-of-box b4 food r1 p4 zone3) [0]
072: ( put-box-on-carrier b4 car2 r1 zone3 s11car2) [0]
073: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
074: ( move-robot-with-carrier r1 car2 zone3 depot) [0]
075: (      put-content-in-box b1 food r1 depot) [0]
076: ( put-box-on-carrier b1 car1 r1 depot s11car1) [0]
077: ( move-robot-with-carrier r1 car1 depot zone4) [0]
078: (drop-box-of-carrier b1 car1 r1 zone4 s11car1) [0]
079: (      drop-content-of-box b1 food r1 p5 zone4) [0]
080: ( put-box-on-carrier b1 car1 r1 zone4 s11car1) [0]
081: (drop-box-of-carrier b4 car2 r2 depot s11car2) [0]
082: ( move-robot-with-carrier r1 car1 zone4 depot) [0]
083: (      put-content-in-box b4 food r1 depot) [0]
084: ( put-box-on-carrier b4 car2 r1 depot s11car2) [0]
085: ( move-robot-with-carrier r1 car2 depot zone5) [0]
086: (drop-box-of-carrier b4 car2 r1 zone5 s11car2) [0]
087: (      drop-content-of-box b4 food r1 p6 zone5) [0]
088: ( put-box-on-carrier b4 car2 r1 zone5 s11car2) [0]
089: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
090: ( move-robot-with-carrier r1 car2 zone5 depot) [0]
091: (      put-content-in-box b1 food r1 depot) [0]
092: ( put-box-on-carrier b1 car1 r1 depot s11car1) [0]
093: ( move-robot-with-carrier r1 car1 depot zone6) [0]
094: (drop-box-of-carrier b1 car1 r1 zone6 s11car1) [0]
095: (      drop-content-of-box b1 food r1 p7 zone6) [0]
096: ( put-box-on-carrier b1 car1 r1 zone6 s11car1) [0]
097: (drop-box-of-carrier b4 car2 r2 depot s11car2) [0]
098: ( move-robot-with-carrier r1 car1 zone6 depot) [0]
099: (      put-content-in-box b4 food r1 depot) [0]
100: ( put-box-on-carrier b4 car2 r1 depot s11car2) [0]
101: ( move-robot-with-carrier r1 car2 depot zone7) [0]
102: (drop-box-of-carrier b4 car2 r1 zone7 s11car2) [0]
103: (      drop-content-of-box b4 food r1 p8 zone7) [0]

```

```

104: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
105: (      put-content-in-box b1 tools r2 depot) [0]
106: ( put-box-on-carrier b1 car1 r2 depot s11car1) [0]
107: ( move-robot-with-carrier r2 car1 depot zone4) [0]
108: (drop-box-of-carrier b1 car1 r2 zone4 s11car1) [0]
109: ( put-box-on-carrier b4 car2 r1 zone7 s11car2) [0]
110: ( move-robot-with-carrier r1 car2 zone7 depot) [0]
111: (      drop-content-of-box b1 tools r2 p5 zone4) [0]
112: (drop-box-of-carrier b4 car2 r1 depot s11car2) [0]
113: (      put-content-in-box b4 tools r1 depot) [0]
114: ( put-box-on-carrier b4 car2 r1 depot s11car2) [0]
115: ( move-robot-with-carrier r1 car2 depot zone5) [0]
116: (drop-box-of-carrier b4 car2 r1 zone5 s11car2) [0]
117: ( put-box-on-carrier b1 car1 r2 zone4 s11car1) [0]
118: ( move-robot-with-carrier r2 car1 zone4 depot) [0]
119: (      drop-content-of-box b4 tools r1 p6 zone5) [0]
120: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
121: (      put-content-in-box b1 tools r2 depot) [0]
122: ( put-box-on-carrier b1 car1 r2 depot s11car1) [0]
123: ( move-robot-with-carrier r2 car1 depot zone6) [0]
124: (drop-box-of-carrier b1 car1 r2 zone6 s11car1) [0]
125: (      drop-content-of-box b1 tools r2 p7 zone6) [0]
126: ( put-box-on-carrier b1 car1 r2 zone6 s11car1) [0]
127: ( move-robot-with-carrier r2 car1 zone6 depot) [0]
128: (drop-box-of-carrier b1 car1 r2 depot s11car1) [0]
129: (      put-content-in-box b1 tools r2 depot) [0]
130: ( put-box-on-carrier b1 car1 r2 depot s11car1) [0]
131: ( move-robot-with-carrier r2 car1 depot zone7) [0]
132: (drop-box-of-carrier b1 car1 r2 zone7 s11car1) [0]
133: (      drop-content-of-box b1 tools r2 p8 zone7) [0]

```

```

time spent:      0.05 seconds parsing
                0.22 seconds encoding
                194.11 seconds searching
                194.38 seconds total time

```

```

memory used:     4.82 MBytes for problem representation
                985.45 MBytes for searching
                990.27 MBytes total

```

## Note aggiuntive sul planner usato

Infine, si specifica che in riferimento al limite in profondità imposto all'algoritmo A\* realizzato in tutte le tre sessioni di esecuzione si è usato un limite ampiamente superiore (200) il quale non ha limitato in alcun modo la ricerca del goal ma solo impedito che si verificasse un trabocco di memoria.

Inoltre, in conclusione di questa sezione, possiamo affermare con certezza che la combinazione di planner ed euristica trovati hanno una buona efficienza soprattutto se confrontati con alcuni già forniti nella libreria *PDDL4J* i quali non riescono nemmeno a portare a termine alcune delle istanze. Chiaramente questo planner non è assolutamente comparabile con quello presente online ma dal momento che riesce risolvere tutte le istanze entro dieci minuti, come richiesto dalla traccia, concludiamo che rispetto alle specifiche fornite il planning ottenuto presenta una buona efficienza.

# Temporal Planning & Robotics

## Temporal Planning

Nella prima parte della traccia si richiede di convertire il dominio e la prima istanza del problema in *durative-actions* in modo da modellare lo stesso problema ma in modo temporale. Inoltre si impone un peso per ciascun contenuto (cibo, medicine e attrezzi), tale peso influenzerà il movimento del carrello dato che, in base al contenuto, il suo peso andrà ad aumentare e avrà un maggior costo quando si dovrà effettuare l'azione impattando direttamente il costo complessivo dell'operazione.

Per effettuare l'operazione di conversione del dominio, come prima cosa, è stato necessario introdurre delle funzioni, denotate dalla dichiarazione *functions*, in modo da definire il costo totale delle operazioni (*path-cost*), il peso del contenuto (*content-cost*), il peso della scatola (*box-cost*) e il peso del carrello (*carrier-cost*).

```
(:functions
  (path-cost ?rob - robot)
  (content-cost ?c - content)
  (box-cost ?b - box)
  (carrier-cost ?car - carrier)
)
```

Successivamente, per tenere conto dello stato del robot, ovvero se occupato in un'operazione oppure libero di agire, è stato aggiunto il predicato *free-robot* che denota il robot come disponibile.

```
(:predicates
  (at ?obj - locatable ?zone - zone)
  (free-box ?box - box)
  (box-filled ?box - box ?cont - content)
  (box-on-carrier ?box - box ?car - carrier)
  (person-has-content ?per - person ?cont - content)
  (person-needs-content ?per - person ?cont - content)
  (free-slot ?slot - slot)
  (slot-of-carrier ?slot - slot ?car - carrier)
  (free-robot ?rob - robot)
)
```

Come ultima operazione al fine di convertire il dominio si sono convertite tutte le azioni in *durative-action* imponendo un costo all'azione e dichiarando gli effetti all'inizio e alla fine di quest'ultima, come ad esempio negare il predicato *free-robot* all'inizio dell'azione e verificarlo alla fine di quest'ultima. Si evidenzia inoltre che non tutti i pesi di tutte le azioni sono costanti, infatti caricare un contenuto di un certo tipo avrà un costo diverso rispetto a caricare un altro contenuto di tipo diverso, oppure spostare un carrello vuoto non avrà lo stesso costo di spostare un carrello pieno di scatole.

Nello specifico, la durata e il costo delle azioni è stata definendo secondo le seguenti linee guida:

1. Caricare e scaricare un contenuto ha durata 1 e costo pari a quello del contenuto spostato.
2. Caricare e scaricare una scatola ha durata 2 e costo pari a quello della scatola
3. Muovere il robot ha durata e costo 3.
4. Muovere il robot con attaccato il carrello ha durata e costo pari al costo del carrello moltiplicato per 3.

Di seguito si riportano le tutte le *durative actions* presenti nel file *rescue\_domain\_conv.pddl* presente in allegato a cui rimandiamo la visione per maggior chiarezza.

```
(:durative-action put-content-in-box
  :parameters (?box - box ?cont - content ?rob - robot ?zone - zone)
)
:duration(= ?duration 1)
:condition (and
  (at start (free-box ?box))
  (over all (at ?cont ?zone))
  (over all (at ?box ?zone))
  (over all (at ?rob ?zone))
  (at start (free-robot ?rob))
)
:effect (and
  (at start (not (free-robot ?rob)))
  (at start (not (free-box ?box)))
  (at end (free-robot ?rob))
  (at end (box-filled ?box ?cont))
  (at end (increase (path-cost ?rob) (content-cost ?cont)))
  (at end (increase
    (box-cost ?box)
```

```

        (content-cost ?cont)))
    )
)

(:durative-action drop-content-of-box
  :parameters (?box - box ?cont - content ?rob - robot ?per - person ?zone -
zone
  )
  :duration(= ?duration 1)
  :condition (and
    (at start (box-filled ?box ?cont))
    (over all (at ?box ?zone))
    (over all (at ?rob ?zone))
    (over all (at ?per ?zone))
    (over all (person-needs-content ?per ?cont))
    (at start (free-robot ?rob))
  )
  :effect (and
    (at start (not (free-robot ?rob)))
    (at start (not (box-filled ?box ?cont)))
    (at start (person-has-content ?per ?cont))
    (at end (free-robot ?rob))
    (at end (free-box ?box))
    (at end (increase (path-cost ?rob) (content-cost ?cont)))
    (at end (not (person-needs-content ?per ?cont)))
    (at end (decrease
      (box-cost ?box)
      (content-cost ?cont)))
  )
)

(:durative-action put-box-on-carrier
  :parameters (?box - box ?car - carrier ?rob - robot ?zone - zone ?slot -
slot
  )
  :duration(= ?duration 2)
  :condition (and
    (at start (at ?box ?zone))
    (over all (at ?car ?zone))
    (over all (at ?rob ?zone))
    (over all (slot-of-carrier ?slot ?car))
    (at start (free-slot ?slot))
    (at start (free-robot ?rob))
  )
  :effect (and
    (at start (not (free-robot ?rob)))
    (at start (not (at ?box ?zone)))
    (at start (not (free-slot ?slot)))
    (at end (box-on-carrier ?box ?car))
  )
)

```



```

        (at end (free-robot ?rob))
        (at end (increase (path-cost ?rob) (box-cost ?box)))
        (at end (increase (carrier-cost ?car) (box-cost ?box)))
    )
)

(:durative-action drop-box-of-carrier
  :parameters (?box - box ?car - carrier ?rob - robot ?zone - zone ?slot -
slot
  )
  :duration(= ?duration 2)
  :condition (and
    (over all (at ?car ?zone))
    (over all (at ?rob ?zone))
    (at start (box-on-carrier ?box ?car))
    (over all (slot-of-carrier ?slot ?car))
    (at start (free-robot ?rob))
  )
  :effect (and
    (at start (not (free-robot ?rob)))
    (at start (not (box-on-carrier ?box ?car)))
    (at end (free-robot ?rob))
    (at end (at ?box ?zone))
    (at end (free-slot ?slot))
    (at end (increase (path-cost ?rob) (box-cost ?box)))
    (at end (decrease (carrier-cost ?car) (box-cost ?box)))
  )
)

(:durative-action move-robot
  :parameters (?rob - robot ?from - zone ?to - zone
  )
  :duration(= ?duration 3)
  :condition (and
    (at start (at ?rob ?from))
    (at start (free-robot ?rob))
  )
  :effect (and
    (at start (not (free-robot ?rob)))
    (at start (not (at ?rob ?from)))
    (at end (free-robot ?rob))
    (at end (at ?rob ?to))
    (at end (increase (path-cost ?rob) 3))
  )
)

(:durative-action move-robot-with-carrier
  :parameters (?rob - robot ?car - carrier ?from - zone ?to - zone
  )
  :duration (= ?duration (* (carrier-cost ?car) 3))
  :condition (and

```

```

        (at start (at ?rob ?from))
        (at start (at ?car ?from))
        (at start (free-robot ?rob))
    )
    :effect (and
        (at start (not (free-robot ?rob)))
        (at start (not (at ?rob ?from)))
        (at start (not (at ?car ?from)))
        (at end (free-robot ?rob))
        (at end (at ?rob ?to))
        (at end (at ?car ?to))
        (at end (increase
            (path-cost ?rob)
            (* (carrier-cost ?car) 3)))
    )
)

```

La prima istanza del problema è stata anch'essa modificata al fine di inglobare le modifiche, impostare i pesi dei contenuti e specificare come obiettivo quello di minimizzare il costo del path risolutivo complessivo. Di seguito si riporta il file *rescue\_instance1\_conv.pddl* contenente quanto specificato.

```

(define (problem rescue_inst_1_conv)
  (:domain rescue_sys_conv)
  (:objects
    depot zone1 zone2 - zone
    b1 b2 b3 b4 b5 - box
    p1 p2 p3 - person
    r - robot
    car - carrier
    sl1 sl2 sl3 sl4 - slot
    drugs food tools - content
  )
  (:init
    (= (content-cost drugs) 1)
    (= (content-cost food) 2)
    (= (content-cost tools) 3)
    (= (box-cost b1) 0)
    (= (box-cost b2) 0)
    (= (box-cost b3) 0)
    (= (box-cost b4) 0)
    (= (box-cost b5) 0)
    (= (carrier-cost car) 1)
    (= (path-cost r) 0)
    (at b1 depot)
    (at b2 depot)
    (at b3 depot)
    (at b4 depot)
    (at b5 depot)
  )
)

```

```

    (at r depot)
    (free-robot r)
    (at car depot)
    (at food depot)
    (at drugs depot)
    (at tools depot)
    (at p1 zone1)
    (at p2 zone1)
    (at p3 zone2)
    (person-needs-content p1 food)
    (person-needs-content p1 drugs)
    (person-needs-content p2 drugs)
    (person-needs-content p3 food)
    (free-box b1)
    (free-box b2)
    (free-box b3)
    (free-box b4)
    (free-box b5)
    (free-slot sl1)
    (free-slot sl2)
    (free-slot sl3)
    (free-slot sl4)
    (slot-of-carrier sl1 car)
    (slot-of-carrier sl2 car)
    (slot-of-carrier sl3 car)
    (slot-of-carrier sl4 car)
  )
  (:goal
    (and
      (person-has-content p1 food)
      (person-has-content p1 drugs)
      (person-has-content p2 drugs)
      (person-has-content p3 food)
    )
  )
  (:metric minimize (path-cost r))
)

```

A questo punto possiamo usare la libreria *planutils* per generare un piano risolutivo al problema temporale appena definito. A questo scopo usiamo il planner *lpg-td* disponibile nella libreria *planutils* richiamandolo con il seguente comando.

```
$ planutils run lpg-td domain.pddl problem.pddl
```

Il piano ottenuto in output dal comando è il seguente.

```
Plan computed:
```

```

Time: (ACTION) [action Duration; action Cost]
0.0003: (PUT-CONTENT-IN-BOX B4 DRUGS R DEPOT) [D:1.0000; C:1.0000]
1.0005: (PUT-BOX-ON-CARRIER B4 CAR R DEPOT SL1) [D:2.0000; C:1.0000]
3.0008: (PUT-CONTENT-IN-BOX B5 DRUGS R DEPOT) [D:1.0000; C:1.0000]
4.0010: (PUT-BOX-ON-CARRIER B5 CAR R DEPOT SL2) [D:2.0000; C:1.0000]
6.0012: (PUT-CONTENT-IN-BOX B2 FOOD R DEPOT) [D:1.0000; C:2.0000]
7.0015: (PUT-BOX-ON-CARRIER B2 CAR R DEPOT SL3) [D:2.0000; C:2.0000]
9.0017: (PUT-CONTENT-IN-BOX B1 FOOD R DEPOT) [D:1.0000; C:2.0000]
10.0020: (PUT-BOX-ON-CARRIER B1 CAR R DEPOT SL4) [D:2.0000; C:2.0000]
12.0022: (MOVE-ROBOT-WITH-CARRIER R CAR DEPOT ZONE1) [D:21.0000; C:21.0000]
33.0025: (DROP-BOX-OF-CARRIER B4 CAR R ZONE1 SL2) [D:2.0000; C:1.0000]
35.0028: (DROP-CONTENT-OF-BOX B4 DRUGS R P1 ZONE1) [D:1.0000; C:1.0000]
36.0030: (DROP-BOX-OF-CARRIER B5 CAR R ZONE1 SL1) [D:2.0000; C:1.0000]
38.0033: (DROP-CONTENT-OF-BOX B5 DRUGS R P2 ZONE1) [D:1.0000; C:1.0000]
39.0035: (DROP-BOX-OF-CARRIER B2 CAR R ZONE1 SL3) [D:2.0000; C:2.0000]
41.0037: (DROP-CONTENT-OF-BOX B2 FOOD R P1 ZONE1) [D:1.0000; C:2.0000]
42.0040: (MOVE-ROBOT-WITH-CARRIER R CAR ZONE1 ZONE2) [D:9.0000; C:9.0000]
51.0042: (DROP-BOX-OF-CARRIER B1 CAR R ZONE2 SL4) [D:2.0000; C:2.0000]
53.0045: (DROP-CONTENT-OF-BOX B1 FOOD R P3 ZONE2) [D:1.0000; C:2.0000]

Solution found:
Total time:      4.01
Search time:     2.03
Actions:         18
Execution cost:  54.00
Duration:        54.000
Plan quality:    54.000

```

## Robotics Planning

Come ultima parte di questo terzo punto viene richiesto di integrare il piano appena generato in una architettura software robotica per il mondo reale, a tale scopo è necessario usare il pacchetto *PlanSys2*. Nello specifico si costruisce un nuovo pacchetto per *PlanSys2* in cui è necessario:

1. Ridefinire le azioni del dominio come *fake-action* (file C++ che costituisce un'azione per *PlanSys2* e che fa riferimento al dominio *PDDL* definito).
2. Modificare il file del dominio in modo da renderlo compatibile con *PlanSys2* rimuovendo la gerarchia dei tipi, le durate delle azioni variabili e il calcolo del costo delle varie operazioni. Questo è necessario dati gli errori incontrati avviando il pacchetto con il dominio definito prima, infatti i requirements *numeric-fluents* e *duration-inequalities* non vengono

riconosciuti rendendo impossibile il prosieguo delle operazioni. Inoltre, tale restrizione è giustificata dal fatto che l'applicazione del piano generato deve rispettare i soli vincoli temporali del dominio modificato per *PlanSys2*.

3. Convertire la prima istanza del problema (condizioni iniziali e goal) in operazioni compatibili con la sintassi del terminale di PlanSys2. Per comodità sono tutti riportati in un unico file in modo da non doverli eseguire uno per volta.
4. Implementare un file python contenente la descrizione di avvio del pacchetto e il mapping alle *fake-actions*.
5. Definire le dipendenze e le operazioni di costruzione del pacchetto in un file di tipo *CMake*.
6. Riportare le informazioni aggiuntive (come maintainer e licenza) del pacchetto che si sta costruendo in un file *package.xml*.

Le *fake-action* definite dai file C++ che mappano le *durative-action* del dominio sono pressoché identiche a meno del nome e dei log che emettono, per tale motivo ne riportiamo una sola (*put\_box\_on\_carrier.cpp*) e rimandiamo alla visione della cartella di progetto *plansys2\_project* presente in allegato il cui contenuto verrà descritto nella sezione [Deliverables](#).

```
#include <memory>
#include <algorithm>
#include "plansys2_executor/ActionExecutorClient.hpp"
#include "rclcpp/rclcpp.hpp"
#include "rclcpp_action/rclcpp_action.hpp"

using namespace std::chrono_literals;

class PutBoxOnCarrier : public plansys2::ActionExecutorClient
{
public:
    PutBoxOnCarrier()
        : plansys2::ActionExecutorClient("put_box_on_carrier", 200ms)
    {
        progress_ = 0.0;
    }

private:
    void do_work()
    {
        std::vector<std::string> arguments = get_arguments();
        if (progress_ < 1.0)
        {
            progress_ += 0.2;
            send_feedback(progress_, "Robot " + arguments[2] + " is putting box
```

```

" + arguments[0] + " on the carrier " + arguments[1] + " in zone " +
arguments[3] + " occupying slot " + arguments[4] + "\n");
    }
    else
    {
        finish(true, 1.0, "Robot " + arguments[2] + " putted box " +
arguments[0] + " on the carrier " + arguments[1] + " in zone " + arguments[3] +
" occupying slot " + arguments[4] + "\n");
        progress_ = 0.0;
        std::cout << std::endl;
    }
    std::cout << "Robot " + arguments[2] + " is putting box " + arguments[0]
+ " on the carrier " + arguments[1] + " in zone " + arguments[3] + " occupying
slot " + arguments[4] + "\n" << std::flush;
}
float progress_;
};

int main(int argc, char **argv)
{
    rclcpp::init(argc, argv);
    auto node = std::make_shared<PutBoxOnCarrier>();
    node->set_parameter(rclcpp::Parameter("action_name", "put_box_on_carrier"));

    node->trigger_transition(lifecycle_msgs::msg::Transition::TRANSITION_CONFIGURE);
    rclcpp::spin(node->get_node_base_interface());
    rclcpp::shutdown();
    return 0;
}

```

Al fine di rimuovere la gerarchia dei tipi dal dominio e avere tipi tutti su uno stesso livello è stato necessario scorporare le relazioni introdotte e introdurre nuovi predicati qualora questi ultimi facessero riferimento a tipi non concreti. In questo caso, è stato necessario ridefinire il predicato *at* introducendo *robot\_at*, *box\_at*, *carrier\_at* e *person\_at*. Come detto in precedenza si sono rimosse anche le operazioni per il calcolo del costo delle operazioni e le durate dipendenti da essi, di seguito si riporta tale dominio modificato presente nel file *rescue\_domain\_plansys.pddl*.

```

(define (domain rescue_sys_plansys)
  (:requirements :negative-preconditions :typing :conditional-effects :adl
:universal-preconditions :durative-actions)

  (:types
    zone locatable slot person box content robot carrier
  )

  (:predicates
    (robot_at ?rob - robot ?zone - zone)
    (box_at ?box - box ?zone - zone)
    (carrier_at ?car - carrier ?zone - zone)
    (content_at ?cont - content ?zone - zone)
  )

```

```

    (person_at ?per - person ?zone - zone)
    (free_box ?box - box)
    (box_filled ?box - box ?cont - content)
    (box_on_carrier ?box - box ?car - carrier)
    (person_has_content ?per - person ?cont - content)
    (person_needs_content ?per - person ?cont - content)
    (free_slot ?slot - slot)
    (slot_of_carrier ?slot - slot ?car - carrier)
    (free_robot ?rob - robot)
  )

  (:durative-action put_content_in_box
    :parameters (?box - box ?cont - content ?rob - robot ?zone - zone
    )
    :duration(= ?duration 1)
    :condition (and
      (at start (free_box ?box))
      (over all (content_at ?cont ?zone))
      (over all (box_at ?box ?zone))
      (over all (robot_at ?rob ?zone))
      (at start (free_robot ?rob))
    )
    :effect (and
      (at start (not (free_robot ?rob)))
      (at start (not (free_box ?box)))
      (at end (free_robot ?rob))
      (at end (box_filled ?box ?cont))
    )
  )

  (:durative-action drop_content_of_box
    :parameters (?box - box ?cont - content ?rob - robot ?per - person ?zone
- zone
    )
    :duration(= ?duration 1)

    :condition (and
      (at start (box_filled ?box ?cont))
      (over all (box_at ?box ?zone))
      (over all (robot_at ?rob ?zone))
      (over all (person_at ?per ?zone))
      (over all (person_needs_content ?per ?cont))
      (at start (free_robot ?rob))
    )
    :effect (and
      (at start (not (free_robot ?rob)))
      (at start (not (box_filled ?box ?cont)))
      (at start (person_has_content ?per ?cont))
      (at end (free_robot ?rob))
      (at end (free_box ?box))
      (at end (not (person_needs_content ?per ?cont)))
    )
  )

  (:durative-action put_box_on_carrier
    :parameters (?box - box ?car - carrier ?rob - robot ?zone - zone ?slot -
slot

```

```

)
:duration(= ?duration 2)
:condition (and
  (at start (box_at ?box ?zone))
  (over all (carrier_at ?car ?zone))
  (over all (robot_at ?rob ?zone))
  (over all (slot_of_carrier ?slot ?car))
  (at start (free_slot ?slot))
  (at start (free_robot ?rob))
)
:effect (and
  (at start (not (free_robot ?rob)))
  (at start (not (box_at ?box ?zone)))
  (at start (not (free_slot ?slot)))
  (at end (box_on_carrier ?box ?car))
  (at end (free_robot ?rob))
)
)

(:durative-action drop_box_of_carrier
  :parameters (?box - box ?car - carrier ?rob - robot ?zone - zone ?slot -
slot
)
  :duration(= ?duration 2)
  :condition (and
    (over all (carrier_at ?car ?zone))
    (over all (robot_at ?rob ?zone))
    (at start (box_on_carrier ?box ?car))
    (over all (slot_of_carrier ?slot ?car))
    (at start (free_robot ?rob))
  )
  :effect (and
    (at start (not (free_robot ?rob)))
    (at start (not (box_on_carrier ?box ?car)))
    (at end (free_robot ?rob))
    (at end (box_at ?box ?zone))
    (at end (free_slot ?slot))
  )
)

(:durative-action move_robot
  :parameters (?rob - robot ?from - zone ?to - zone
)
  :duration(= ?duration 3)
  :condition (and
    (at start (robot_at ?rob ?from))
    (at start (free_robot ?rob))
  )
  :effect (and
    (at start (not (free_robot ?rob)))
    (at start (not (robot_at ?rob ?from)))
    (at end (free_robot ?rob))
    (at end (robot_at ?rob ?to))
  )
)

(:durative-action move_robot_with_carrier
  :parameters (?rob - robot ?car - carrier ?from - zone ?to - zone

```



```

    )
    :duration(= ?duration 3)
    :condition (and
      (at start (robot_at ?rob ?from))
      (at start (carrier_at ?car ?from))
      (at start (free_robot ?rob))
    )
    :effect (and
      (at start (not (free_robot ?rob)))
      (at start (not (robot_at ?rob ?from)))
      (at start (not (carrier_at ?car ?from)))
      (at end (free_robot ?rob))
      (at end (robot_at ?rob ?to))
      (at end (carrier_at ?car ?to))
    )
  )
)
)

```

Tali modifiche impattano anche le condizioni iniziali della prima istanza del problema (quella cui si riferisce il piano generato nella sezione precedente) e per tale motivo è stato necessario applicare delle modifiche in tal senso per ridefinire le condizioni iniziali. Di seguito si riporta il file commands che include tali modifiche e le rende compatibili con il terminale di *PlanSys2*.

```

set instance depot zone
set instance zone1 zone
set instance zone2 zone
set instance b1 box
set instance b2 box
set instance b3 box
set instance b4 box
set instance b5 box
set instance p1 person
set instance p2 person
set instance p3 person
set instance r robot
set instance car carrier
set instance s11 slot
set instance s12 slot
set instance s13 slot
set instance s14 slot
set instance drugs content
set instance food content
set instance tools content

set predicate (box_at b1 depot)
set predicate (box_at b2 depot)
set predicate (box_at b3 depot)
set predicate (box_at b4 depot)
set predicate (box_at b5 depot)

set predicate (robot_at r depot)
set predicate (free_robot r)
set predicate (carrier_at car depot)

```

```

set predicate (content_at food depot)
set predicate (content_at drugs depot)
set predicate (content_at tools depot)

set predicate (person_at p1 zone1)
set predicate (person_at p2 zone1)
set predicate (person_at p3 zone2)

set predicate (person_needs_content p1 food)
set predicate (person_needs_content p1 drugs)
set predicate (person_needs_content p2 drugs)
set predicate (person_needs_content p3 food)

set predicate (free_box b1)
set predicate (free_box b2)
set predicate (free_box b3)
set predicate (free_box b4)
set predicate (free_box b5)

set predicate (free_slot s11)
set predicate (free_slot s12)
set predicate (free_slot s13)
set predicate (free_slot s14)

set predicate (slot_of_carrier s11 car)
set predicate (slot_of_carrier s12 car)
set predicate (slot_of_carrier s13 car)
set predicate (slot_of_carrier s14 car)

set goal (and (person-has-content p1 food) (person-has-content p1 drugs)
(person-has-content p2 drugs) (person-has-content p3 food))

```

Per eseguire il nuovo pacchetto, come detto in precedenza, è necessario definire una configurazione di lancio in un file Python nel quale si mappano le *fake-action* e si imposta la referenza al dominio PDDL specifico. Di seguito si riporta tale file denominato *plansys2\_project\_launch.py*.

```

import os
from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument, IncludeLaunchDescription
from launch.launch_description_sources import PythonLaunchDescriptionSource
from launch.substitutions import LaunchConfiguration
from launch_ros.actions import Node

def generate_launch_description():
    # Get the launch directory
    example_dir = get_package_share_directory("plansys2_project")
    namespace = LaunchConfiguration("namespace")
    declare_namespace_cmd = DeclareLaunchArgument(
        "namespace", default_value="", description="Namespace"
    )

```

```

)
plansys2_cmd = IncludeLaunchDescription(
    PythonLaunchDescriptionSource(
        os.path.join(
            get_package_share_directory("plansys2_bringup"),
            "launch",
            "plansys2_bringup_launch_monolithic.py",
        )
    ),
    launch_arguments={
        "model_file": example_dir + "/pddl/rescue_domain_plansys.pddl",
        "namespace": namespace,
    }.items(),
)
# Specify the actions
drop_box_of_carrier_cmd = Node(
    package='plansys2_project',
    executable='drop_box_of_carrier_node',
    name='drop_box_of_carrier_node',
    namespace=namespace,
    output='screen',
    parameters=[])
drop_content_of_box_cmd = Node(
    package='plansys2_project',
    executable='drop_content_of_box_node',
    name='drop_content_of_box_node',
    namespace=namespace,
    output='screen',
    parameters=[])
move_robot_with_carrier_cmd = Node(
    package='plansys2_project',
    executable='move_robot_with_carrier_node',
    name='move_robot_with_carrier_node',
    namespace=namespace,
    output='screen',
    parameters=[])
move_robot_cmd = Node(
    package='plansys2_project',
    executable='move_robot_node',
    name='move_robot_node',
    namespace=namespace,
    output='screen',
    parameters=[])
put_box_on_carrier_cmd = Node(
    package='plansys2_project',
    executable='put_box_on_carrier_node',
    name='put_box_on_carrier_node',
    namespace=namespace,
    output='screen',
    parameters=[])
put_content_in_box_cmd = Node(

```

```

package='plansys2_project',
executable='put_content_in_box_node',
name='put_content_in_box_node',
namespace=namespace,
output='screen',
parameters=[])
# Create the launch description and populate
ld = LaunchDescription()
ld.add_action(declare_namespace_cmd)
ld.add_action(plansys2_cmd)
ld.add_action(drop_box_of_carrier_cmd)
ld.add_action(drop_content_of_box_cmd)
ld.add_action(move_robot_with_carrier_cmd)
ld.add_action(move_robot_cmd)
ld.add_action(put_box_on_carrier_cmd)
ld.add_action(put_content_in_box_cmd)
return ld

```

La gestione delle dipendenze e della configurazione di build è gestita mediante un file di tipo *CMake* denominato *CMakeLists.txt* e riportato di seguito.

```

cmake_minimum_required(VERSION 3.5)
project(plansys2_project)

find_package(ament_cmake REQUIRED)
find_package(rclcpp REQUIRED)
find_package(rclcpp_action REQUIRED)
find_package(plansys2_msgs REQUIRED)
find_package(plansys2_executor REQUIRED)

set(CMAKE_CXX_STANDARD 17)
set(dependencies
  rclcpp
  rclcpp_action
  plansys2_msgs
  plansys2_executor
)

add_executable(drop_box_of_carrier_node src/drop_box_of_carrier.cpp)
ament_target_dependencies(drop_box_of_carrier_node ${dependencies})
add_executable(drop_content_of_box_node src/drop_content_of_box.cpp)
ament_target_dependencies(drop_content_of_box_node ${dependencies})
add_executable(move_robot_with_carrier_node src/move_robot_with_carrier.cpp)
ament_target_dependencies(move_robot_with_carrier_node ${dependencies})
add_executable(move_robot_node src/move_robot.cpp)
ament_target_dependencies(move_robot_node ${dependencies})
add_executable(put_box_on_carrier_node src/put_box_on_carrier.cpp)
ament_target_dependencies(put_box_on_carrier_node ${dependencies})
add_executable(put_content_in_box_node src/put_content_in_box.cpp)

```

```

ament_target_dependencies(put_content_in_box_node ${dependencies})

install(DIRECTORY launch pddl DESTINATION share/${PROJECT_NAME})
install(TARGETS
  drop_box_of_carrier_node
  drop_content_of_box_node
  move_robot_with_carrier_node
  move_robot_node
  put_box_on_carrier_node
  put_content_in_box_node
  ARCHIVE DESTINATION lib
  LIBRARY DESTINATION lib
  RUNTIME DESTINATION lib/${PROJECT_NAME}
)

if(BUILD_TESTING)
  find_package(ament_lint_auto REQUIRED)
  ament_lint_auto_find_test_dependencies()
  find_package(ament_cmake_gtest REQUIRED)
endif()

ament_export_dependencies(${dependencies})
ament_package()

```

Per concludere la descrizione dei file che compongono il pacchetto si riporta anche il file *package.xml*.

```

<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd"
schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
  <name>plansys2_project</name>
  <version>0.0.4</version>

  <description>ROS2 Planning System</description>

  <maintainer email="brunoguzzo18@gmail.com">Bruno Guzzo</maintainer>

  <license>Apache License, Version 2.0</license>

  <buildtool_depend>ament_cmake</buildtool_depend>

  <depend>rclcpp</depend>
  <depend>rclcpp_action</depend>
  <depend>plansys2_msgs</depend>
  <depend>plansys2_executor</depend>

  <exec_depend>plansys2_bringup</exec_depend>
  <exec_depend>plansys2_terminal</exec_depend>

```

```

<test_depend>ament_lint_common</test_depend>
<test_depend>ament_lint_auto</test_depend>
<test_depend>ament_cmake_gtest</test_depend>

<export>
  <build_type>ament_cmake</build_type>
</export>
</package>

```

Per compilare il pacchetto è necessario eseguire i seguenti comandi nella cartella radice del progetto:

```

~/ia-project/plansys2_project$ colcon build --symlink-install
~/ia-project/plansys2_project$ source install/setup.bash

```

Successivamente è necessario aprire due terminali e nel primo si lancia il pacchetto con lo script di lancio Python descritto prima.

```

~/ia-project/plansys2_project$ ros2 launch plansys2_project
plansys2_project_launch.py

```

Se il dominio PDDL è stato correttamente interpretato non saranno presenti errori e tutti i nodi verranno avviati in modo corretto ricevendo a video il seguente output.

```

[INFO] [launch]: All log files can be found below
/home/bruno/.ros/log/2023-09-05-07-42-43-386313-bruno-VM-2252
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [plansys2_node-1]: process started with pid [2256]
[INFO] [drop_box_of_carrier_node-2]: process started with pid [2258]
[INFO] [drop_content_of_box_node-3]: process started with pid [2260]
[INFO] [move_robot_with_carrier_node-4]: process started with pid [2262]
[INFO] [move_robot_node-5]: process started with pid [2264]
[INFO] [put_box_on_carrier_node-6]: process started with pid [2266]
[INFO] [put_content_in_box_node-7]: process started with pid [2268]
[plansys2_node-1] [INFO] [1693892564.240728772] [domain_expert_lc_mgr]:
Creating client for service [domain_expert/get_state]
[plansys2_node-1] [INFO] [1693892564.240884161] [domain_expert_lc_mgr]:
Creating client for service [domain_expert/change_state]
[plansys2_node-1] [INFO] [1693892564.242695021] [executor_lc_mgr]: Creating
client for service [executor/get_state]
[plansys2_node-1] [INFO] [1693892564.242735467] [executor_lc_mgr]: Creating
client for service [executor/change_state]
[plansys2_node-1] [INFO] [1693892564.245046092] [planner_lc_mgr]: Creating
client for service [planner/get_state]
[plansys2_node-1] [INFO] [1693892564.245088914] [planner_lc_mgr]: Creating

```

```

client for service [planner/change_state]
[plansys2_node-1] [INFO] [1693892564.246959505] [problem_expert_lc_mgr]:
Creating client for service [problem_expert/get_state]
[plansys2_node-1] [INFO] [1693892564.247001485] [problem_expert_lc_mgr]:
Creating client for service [problem_expert/change_state]
[plansys2_node-1] [INFO] [1693892564.249320901] [domain_expert]: [domain_expert]
Configuring...
[plansys2_node-1] [INFO] [1693892564.486396959] [domain_expert]: [domain_expert]
Configured
[plansys2_node-1] [INFO] [1693892564.486862354] [domain_expert_lc_mgr]:
Transition 1 successfully triggered.
[plansys2_node-1] [INFO] [1693892564.487301256] [domain_expert_lc_mgr]: Node
domain_expert_lc_mgr has current state inactive.
[plansys2_node-1] [INFO] [1693892564.487482637] [problem_expert]:
[problem_expert] Configuring...
[plansys2_node-1] [INFO] [1693892564.488048771] [problem_expert]:
[problem_expert] Configured
[plansys2_node-1] [INFO] [1693892564.488240486] [problem_expert_lc_mgr]:
Transition 1 successfully triggered.
[plansys2_node-1] [INFO] [1693892564.488576464] [problem_expert_lc_mgr]: Node
problem_expert_lc_mgr has current state inactive.
[plansys2_node-1] [INFO] [1693892564.488742638] [planner]: [planner]
Configuring...
[plansys2_node-1] [INFO] [1693892564.489578262] [planner]: Created solver : POPF
of type plansys2/POPFPlanSolver
[plansys2_node-1] [INFO] [1693892564.489624963] [planner]: [planner] Configured
[plansys2_node-1] [INFO] [1693892564.489845557] [planner_lc_mgr]: Transition 1
successfully triggered.
[plansys2_node-1] [INFO] [1693892564.490140517] [planner_lc_mgr]: Node
planner_lc_mgr has current state inactive.
[plansys2_node-1] [INFO] [1693892564.490300166] [executor]: [executor]
Configuring...
[plansys2_node-1] [INFO] [1693892564.558981901] [executor]: [executor]
Configured
[plansys2_node-1] [INFO] [1693892564.559302893] [executor_lc_mgr]: Transition 1
successfully triggered.
[plansys2_node-1] [INFO] [1693892564.559714721] [executor_lc_mgr]: Node
executor_lc_mgr has current state inactive.
[plansys2_node-1] [INFO] [1693892564.559894568] [domain_expert]: [domain_expert]
Activating...
[plansys2_node-1] [INFO] [1693892564.559910816] [domain_expert]: [domain_expert]
Activated
[plansys2_node-1] [INFO] [1693892564.560039311] [domain_expert_lc_mgr]:
Transition 3 successfully triggered.
[plansys2_node-1] [INFO] [1693892564.560165792] [problem_expert]:
[problem_expert] Activating...
[plansys2_node-1] [INFO] [1693892564.560182542] [problem_expert]:
[problem_expert] Activated
[plansys2_node-1] [INFO] [1693892564.560390686] [problem_expert_lc_mgr]:
Transition 3 successfully triggered.
[plansys2_node-1] [INFO] [1693892564.560510219] [planner]: [planner]

```

```

Activating...
[plansys2_node-1] [INFO] [1693892564.560525015] [planner]: [planner] Activated
[plansys2_node-1] [INFO] [1693892564.560640138] [planner_lc_mgr]: Transition 3
successfully triggered.
[plansys2_node-1] [INFO] [1693892564.560754900] [executor]: [executor]
Activating...
[plansys2_node-1] [INFO] [1693892564.560769194] [executor]: [executor] Activated
[plansys2_node-1] [INFO] [1693892564.560877301] [executor_lc_mgr]: Transition 3
successfully triggered.
[plansys2_node-1] [INFO] [1693892564.561096351] [domain_expert_lc_mgr]: Node
domain_expert_lc_mgr has current state active.
[plansys2_node-1] [INFO] [1693892564.561264751] [problem_expert_lc_mgr]: Node
problem_expert_lc_mgr has current state active.
[plansys2_node-1] [INFO] [1693892564.561426655] [planner_lc_mgr]: Node
planner_lc_mgr has current state active.
[plansys2_node-1] [INFO] [1693892564.561600899] [executor_lc_mgr]: Node
executor_lc_mgr has current state active.

```

Nel secondo terminale, invece, si esegue il seguente comando per accedere al terminale di PlanSys2.

```
~/ia-project/plansys2_project$ ros2 run plansys2_terminal plansys2_terminal
```

A questo punto possiamo impostare le condizioni iniziali dando in ingresso il file *commands* con il seguente comando.

```
> source
/home/bruno/Documents/GitHub/ia-project/plansys2_project/launch/commands
```

Se tutto è andato a buon fine il comando non restituirà output segnalando che i predicati e i goal sono stati correttamente acquisiti e verificati. Ora non rimane che eseguire il piano generato nella sezione precedente al quale sono state effettuate delle leggere modifiche per renderlo conforme alla sintassi utilizzata nel progetto *PlanSys2* come ad esempio trasformarlo in lowercase e lasciare solo la durata dell'azione a destra tra parentesi quadre. A ogni modo tale piano è presente nel file *rescue\_plan.plan* riportato di seguito e presente in allegato.

```

0.0003: (put_content_in_box b4 drugs r depot) [1]
1.0005: (put_box_on_carrier b4 car r depot s11) [2]
3.0008: (put_content_in_box b5 drugs r depot) [1]
4.0010: (put_box_on_carrier b5 car r depot s12) [2]
6.0012: (put_content_in_box b2 food r depot) [1]
7.0015: (put_box_on_carrier b2 car r depot s13) [2]
9.0017: (put_content_in_box b1 food r depot) [1]
10.0020: (put_box_on_carrier b1 car r depot s14) [2]

```



```

12.0022: (move_robot_with_carrier r car depot zone1) [21]
33.0025: (drop_box_of_carrier b4 car r zone1 sl2) [2]
35.0028: (drop_content_of_box b4 drugs r p1 zone1) [1]
36.0030: (drop_box_of_carrier b5 car r zone1 sl1) [2]
38.0033: (drop_content_of_box b5 drugs r p2 zone1) [1]
39.0035: (drop_box_of_carrier b2 car r zone1 sl3) [2]
41.0037: (drop_content_of_box b2 food r p1 zone1) [1]
42.0040: (move_robot_with_carrier r car zone1 zone2) [9]
51.0042: (drop_box_of_carrier b1 car r zone2 sl4) [2]
53.0045: (drop_content_of_box b1 food r p3 zone2) [1]

```

Per eseguire tale piano è sufficiente eseguire il seguente comando nel secondo terminale, quello in cui è aperto a sua volta un terminale di *PlanSys2*.

```

~/ia-project/plansys2_project$ run plan-file
/home/bruno/Documents/GitHub/ia-project/plansys2_project/pddl/rescue_plan.plan

```

A questo punto nel primo terminale è possibile osservare l'avanzamento dell'applicazione del piano mentre nel secondo, se tutto è andato a buon fine si restituirà un messaggio di successo come il seguente.

```

[INFO] [1693842567.077334906] [terminal]: No problem file specified.
ROS2 Planning System console. Type "quit" to finish
> source
/home/bruno/Documents/GitHub/ia-project/plansys2_project/launch/commands
> run plan-file
/home/bruno/Documents/GitHub/ia-project/plansys2_project/pddl/rescue_plan.plan
The plan read from
"/home/bruno/Documents/GitHub/ia-project/plansys2_project/pddl/rescue_plan.plan"
is
0.0003:      (put_content_in_box b4 drugs r depot)   [1]
1.0005:      (put_box_on_carrier b4 car r depot sl1) [2]
3.0008:      (put_content_in_box b5 drugs r depot)   [1]
4.001: (put_box_on_carrier b5 car r depot sl2) [2]
6.0012:      (put_content_in_box b2 food r depot)   [1]
7.0015:      (put_box_on_carrier b2 car r depot sl3) [2]
9.0017:      (put_content_in_box b1 food r depot)   [1]
10.002:      (put_box_on_carrier b1 car r depot sl4) [2]
12.0022:     (move_robot_with_carrier r car depot zone1) [21]
33.0025:     (drop_box_of_carrier b4 car r zone1 sl2) [2]
35.0028:     (drop_content_of_box b4 drugs r p1 zone1) [1]
36.003:      (drop_box_of_carrier b5 car r zone1 sl1) [2]
38.0033:     (drop_content_of_box b5 drugs r p2 zone1) [1]
39.0035:     (drop_box_of_carrier b2 car r zone1 sl3) [2]
41.0037:     (drop_content_of_box b2 food r p1 zone1) [1]
42.004:      (move_robot_with_carrier r car zone1 zone2) [9]
51.0042:     (drop_box_of_carrier b1 car r zone2 sl4) [2]
53.0045:     (drop_content_of_box b1 food r p3 zone2) [1]

```

```
[INFO] [1693842606.616841620] [executor_client]: Plan Succeeded
```

```
Successful finished
```

Dal momento che l'output sul primo terminale è estremamente verboso ne riportiamo una versione semplificata ma comunque utile a capire l'applicazione e l'avanzamento delle *fake-action* al fine di raggiungere l'obiettivo.

```
[put_content_in_box_node-7] Robot r is putting content drugs in box b4 in zone depot
[put_box_on_carrier_node-6] Robot r is putting box b4 on the carrier car in zone depot occupying slot s11
[put_content_in_box_node-7] Robot r is putting content drugs in box b5 in zone depot
[put_box_on_carrier_node-6] Robot r is putting box b5 on the carrier car in zone depot occupying slot s12
[put_content_in_box_node-7] Robot r is putting content food in box b2 in zone depot
[put_box_on_carrier_node-6] Robot r is putting box b2 on the carrier car in zone depot occupying slot s13
[put_content_in_box_node-7] Robot r is putting content food in box b1 in zone depot
[put_box_on_carrier_node-6] Robot r is putting box b1 on the carrier car in zone depot occupying slot s14
[move_robot_with_carrier_node-4] Robot r is moving with carrier car from zone depot to zone zone1
[drop_box_of_carrier_node-2] Robot r is dropping box b4 of the carrier car in zone zone1 releasing slot s12
[drop_content_of_box_node-3] Robot r is dropping content drugs of the box b4 in zone p1
[drop_box_of_carrier_node-2] Robot r is dropping box b5 of the carrier car in zone zone1 releasing slot s11
[drop_content_of_box_node-3] Robot r is dropping content drugs of the box b5 in zone p2
[drop_box_of_carrier_node-2] Robot r is dropping box b2 of the carrier car in zone zone1 releasing slot s13
[drop_content_of_box_node-3] Robot r is dropping content food of the box b2 in zone p1
[move_robot_with_carrier_node-4] Robot r is moving with carrier car from zone zone1 to zone zone2
[drop_box_of_carrier_node-2] Robot r is dropping box b1 of the carrier car in zone zone2 releasing slot s14
[drop_content_of_box_node-3] Robot r is dropping content food of the box b1 in zone p3
[plansys2_node-1] [INFO] [1693842604.443263686] [executor]: Plan Succeeded
```

Possiamo concludere questa sezione osservando che il piano viene applicato in modo corretto e rispetta i vincoli temporali imposti nonostante le modifiche

applicate al dominio, infatti l'unica *durative-action* che dipende dal peso dell'azione stessa è quella che sposta il robot con il carrello che nel precedente dominio aveva durata pari al prodotto tra la durata dell'azione di spostamento del robot (3) e il peso del carrello. A seguito delle modifiche sopra descritte l'azione di spostamento del carrello con il robot ha assunto durata uguale a quello del solo spostamento del robot e per questo motivo non sorgono problemi nell'applicazione del piano precedente al dominio e al problema modificati inseriti nel pacchetto *plansys2\_project*.

# Deliverables

Il file zip allegato alla presente relazione non è altro che la [repository](#) GitHub usata in fase di sviluppo per gestire la collaborazione e l'avanzamento del progetto. La struttura di tale repository è la seguente.

```
ia-project/
├── Pddl4jPlanner
│   ├── bin
│   │   └── unical
│   │       ├── AstarCustom$1.class
│   │       ├── AstarCustom.class
│   │       ├── Node.class
│   │       └── PositiveAdjustedSum.class
│   ├── lib
│   │   ├── lombok.jar
│   │   └── pddl4j-4.0.0.jar
│   ├── rescue_domain.pddl
│   ├── rescue_instance1.pddl
│   ├── rescue_instance2.pddl
│   ├── rescue_instance3.pddl
│   └── src
│       └── unical
│           ├── AstarCustom.java
│           ├── Node.java
│           └── PositiveAdjustedSum.java
├── pddl_files
│   ├── rescue_domain.pddl
│   ├── rescue_instance1.pddl
│   ├── rescue_instance2.pddl
│   └── rescue_instance3.pddl
├── plansys2_project
│   ├── build_info.txt
│   ├── CMakeLists.txt
│   ├── install_info.txt
│   ├── launch
│   │   ├── commands
│   │   └── plansys2_project_launch.py
│   ├── output.txt
│   ├── package.xml
│   ├── pddl
│   │   ├── rescue_domain_plansys.pddl
│   │   └── rescue_plan.plan
│   └── src
│       ├── drop_box_of_carrier.cpp
│       ├── drop_content_of_box.cpp
│       ├── move_robot.cpp
│       ├── move_robot_with_carrier.cpp
│       ├── put_box_on_carrier.cpp
│       └── put_content_in_box.cpp
└── planutils
    ├── Dockerfile
    ├── info.txt
    ├── pddl_files_conv
    └── rescue_domain_conv.pddl
```

```
| rescue_instance1_conv.pddl
| README.md
```

La struttura e il significato di tutte le cartelle è riportate di seguito:

1. ***pddl\_files***: Contiene il dominio e le istanze dei problemi per la realizzazione del punto uno e due.
2. ***Pddl4jPlanner***: Progetto Eclipse nel quale si usa la libreria PDDL4J e si definiscono un planner (*AstarCustom.java*) ed un cerusica (*PositiveAdjustedSum.java*) per la realizzazione implementativa del punto due. Per l'esecuzione è necessario importare il progetto Eclipse ed eseguire la classe *AstarCustom* passando per argomento il dominio, il file di problema e il timeout specificato con l'attributo `-t` specificando il tempo in millisecondi. Ad esempio: `rescue_domain.pddl rescue_instance3.pddl -t 999999`
3. ***plansys2\_project***: workspace del progetto *PlanSys2*, contiene tutti i file e le informazioni per la costruzione e l'esecuzione del pacchetto incluso il dominio modificato per ragioni di compatibilità. Questa cartella è stata usata per la realizzazione della seconda parte del terzo punto.
4. ***planutils***: Cartella che contiene istruzioni, dominio e istanza per la realizzazione della prima parte del terzo punto.