

# Progetto di Intelligenza Artificiale

---

Antonella Mercuri, Mat. 247059  
Pasquale Oliveti, Mat. 247275  
Francesco Romeo, Mat. 242584

Luglio 2023

## 1 ARCHIVE STRUCTURE

L'elaborato consegnato consiste in questa relazione e nei file prodotti, organizzati in 3 cartelle:

- La cartella **PDDL-files** contiene tutti i sorgenti PDDL. È suddivisa a sua volta in:
  - **Versione\_1.2**, contenente dominio e problemi, relativi al primo e al secondo punto della traccia.
  - **Versione\_2.1**, contenente dominio e problemi in versione convertita (maggiori dettagli a 4.1) e in versione con i pesi, come richiesto dal terzo punto della traccia.
- La cartella **Planner** contiene i codici sorgenti delle due versioni del planner implementato.
- Il workspace **plansys2\_ws** contiene tutto il necessario per l'esecuzione del problema PlanSys2.

Le cartelle Planner e plansys2\_ws contengono inoltre due file di testo con le rispettive istruzioni per l'esecuzione.

## 2 MODELING

### 2.1 Richieste progettuali

Il primo punto del progetto richiede di modellare, usando **PDDL 1.2**, uno scenario ispirato alla consegna di beni in una situazione di emergenza. In tale scenario è necessario

coordinare degli agenti robotici nella consegna di scatole contenenti i beni appropriati a delle persone ferite presenti in posizioni note. Gli agenti robotici, i carrier su cui caricare le scatole, le scatole ed i beni da consegnare sono tutti inizialmente localizzati ad un “deposito”. Si assume che i beni da consegnare siano sempre sufficienti a soddisfare le necessità dei feriti.

La traccia fornisce inoltre ulteriori indicazioni sulla modellazione dello scenario, tra cui:

- Ad una stessa location possono essere presenti più persone.
- La capacità dei carrier su cui caricare le scatole da consegnare è limitata, ma relativa al singolo problema.
- Gli agenti robotici possono riempire e svuotare scatole, caricare o scaricare scatole da un carrier e spostarsi con o senza carrier.
- Non vi sono percorsi prestabiliti tra le varie posizioni, i robot possono muoversi tra di esse in maniera arbitraria.

## 2.2 Tipi implementati

Alla luce delle richieste progettuali si è deciso di adottare la seguente gerarchia di tipi:

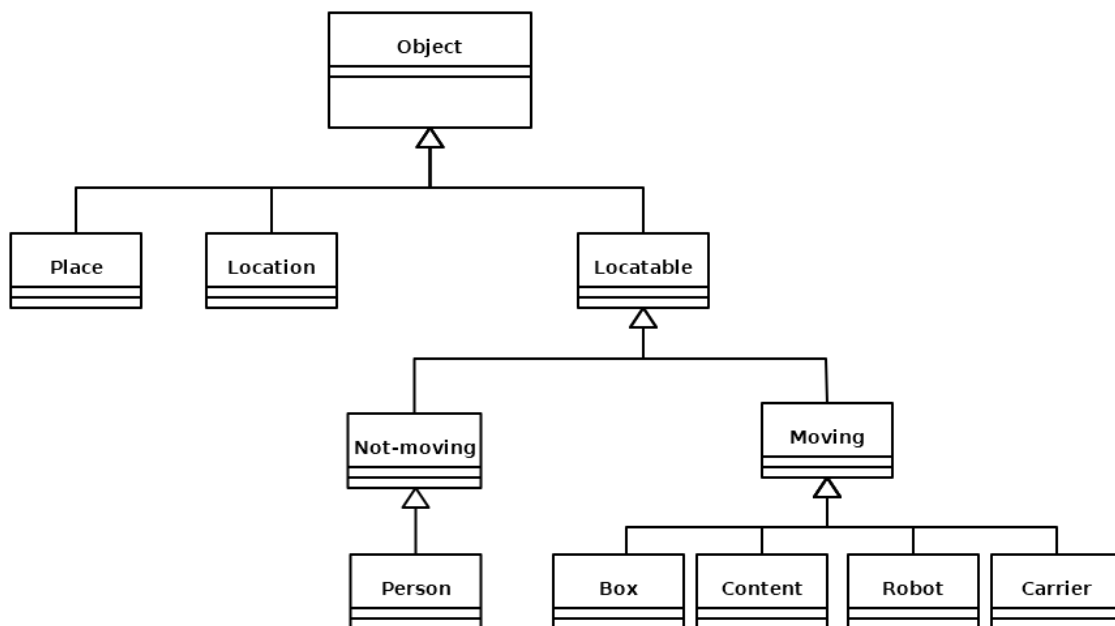


Figura 2.1: Gerarchia dei tipi

In particolare:

- Gli oggetti di tipo **Place** sono dei “token virtuali” utilizzati per modellare i posti disponibili sui carrier.
- Gli oggetti di tipo **Location** rappresentano le posizioni in cui possono trovarsi gli altri oggetti del dominio.
- Tutti gli oggetti “concreti” si trovano in una data posizione, pertanto sono di tipo **Locatable**. Nello specifico gli oggetti Locatable si suddividono in:

- **Not-moving**, cioè gli oggetti che non possono cambiare posizione, di cui fanno parte le sole persone ferite, rappresentate con il tipo **Person**.
- **Moving**, cioè gli oggetti che possono cambiare posizione, di cui fanno parte **Box**, **Content**, **Robot**, **Carrier**. Come richiesto dalla traccia i beni da consegnare sono modellati in maniera generica mediante il tipo **Content** e saranno specificati nei singoli file di problema.

## 2.3 Implementazione in PDDL

Il risultato dello svolgimento del primo punto è il file **domain.pddl**, che modella il dominio del problema mediante tipi, predicati e azioni.

### 2.3.1 Types

Nella sezione **types** sono stati definiti i tipi come discusso nella sezione 2.2.

Listing 1: PDDL types

```
(:types
  location locatable place – object
  moving not-moving – locatable
  person – not-moving
  box content robot carrier – moving
)
```

### 2.3.2 Predicates

Sono stati definiti diversi predicati, in particolare:

- **at** realizza il meccanismo di localizzazione di un locatable. In fase di progettazione si è deciso di localizzare un oggetto solo se non contenuto in altri (ad es. i box caricati sul carrier non si localizzano, ma sfruttano la posizione del carrier). Inoltre non si localizzano nemmeno i beni consegnati alle persone, che vengono considerati “consumati”.
- **box-empty** indica che un box è vuoto.
- **box-contains** indica che un box contiene un certo content.
- **box-loaded-on-carrier** indica che un box è stato caricato su un carrier.
- **person-has-content** indica che una persona ha ricevuto un dato content, viene utilizzato nella definizione del goal.
- **person-needs-content** indica che una persona necessita di un dato content, è il predicato che “guida” la ricerca.
- **place-available** indica che un dato place (token) è disponibile.
- **place-of-carrier** lega un place al carrier a cui fa riferimento.

Listing 2: PDDL predicates

```
(:predicates
  (at ?obj – locatable ?loc – location)
```

```

(box-empty ?box - box)
(box-contains ?box - box ?c - content)

(box-loaded-on-carrier ?box - box ?car - carrier)

(person-has-content ?p - person ?c - content)
(person-needs-content ?p - person ?c - content)

(place-available ?pl - place)
(place-of-carrier ?pl - place ?car - carrier)
)

```

### 2.3.3 Actions

Sono infine state modellate le azioni che l'agente robotico è in grado di compiere:

- **fill-box-with-content:** se il box è vuoto e si trova nella stessa location di robot e content, viene riempito con quel content.

```

(:action fill-box-with-content
  :parameters
    (?b - box
     ?c - content
     ?r - robot
     ?loc - location
    )

  :precondition
    (and
      (box-empty ?b)
      (at ?c ?loc)
      (at ?b ?loc)
      (at ?r ?loc)
    )

  :effect
    (and
      (box-contains ?b ?c)
      (not (box-empty ?b))
    )
)

```

- **empty-box-leaving-content:** se box e robot sono alla stessa location, il box contiene un dato content e a quella location c'è una persona che ha bisogno di quel content, viene effettuata la consegna svuotando il box.

```

(:action empty-box-leaving-content
  :parameters
    (?b - box

```

```

        ?c – content
        ?r – robot
        ?p – person
        ?loc – location
    )

    :precondition
    (and
        (box-contains ?b ?c)
        (at ?b ?loc)
        (at ?r ?loc)
        (at ?p ?loc)
        (person-needs-content ?p ?c)
    )

    :effect
    (and
        (not (box-contains ?b ?c))
        (box-empty ?b)
        (person-has-content ?p ?c)
    )
)

```

- **load-one-box-on-carrier:** se box, carrier e robot sono tutti alla stessa location e c'è un place disponibile sul carrier si carica il box consumando il place (e rimuovendo la posizione del box).

```

(:action load-one-box-on-carrier
  :parameters
  (?b – box
   ?car – carrier
   ?r – robot
   ?loc – location
   ?pl – place
  )

  :precondition
  (and
    (at ?b ?loc)
    (at ?car ?loc)
    (at ?r ?loc)
    (place-of-carrier ?pl ?car)
    (place-available ?pl)
  )

  :effect
  (and
    (not (at ?b ?loc))

```

```

        (box-loaded-on-carrier ?b ?car)
        (not (place-available ?pl))
    )
)

```

- **unload-one-box-from-carrier:** se carrier e robot sono alla stessa location e un box è caricato sul carrier, si scarica tale box assegnandogli la posizione corrente e liberando uno dei place occupati relativi al carrier.

```

(:action unload-one-box-from-carrier
  :parameters
    (?b - box
     ?car - carrier
     ?r - robot
     ?loc - location
     ?pl - place
    )

  :precondition
    (and
      (at ?car ?loc)
      (at ?r ?loc)
      (box-loaded-on-carrier ?b ?car)
      (place-of-carrier ?pl ?car)
      (not (place-available ?pl))
    )

  :effect
    (and
      (at ?b ?loc)
      (not (box-loaded-on-carrier ?b ?car))
      (place-available ?pl)
    )
)

```

- **move-robot:** il robot (da solo) si sposta da una location a un'altra.

```

(:action move-robot
  :parameters
    (?r - robot
     ?from - location
     ?to - location
    )

  :precondition
    (at ?r ?from)

  :effect
    (and

```

```

        (not (at ?r ?from))
        (at ?r ?to)
    )
)

```

- **move-robot-carrier**: il robot si sposta con il carrier da una location a un'altra.

```

(:action move-robot-carrier
  :parameters
    (?r - robot
     ?car - carrier
     ?from - location
     ?to - location
    )

  :precondition
    (and
      (at ?r ?from)
      (at ?car ?from)
    )

  :effect
    (and
      (not (at ?r ?from))
      (at ?r ?to)
      (not (at ?car ?from))
      (at ?car ?to)
    )
)

```

## 2.4 Testing

Il testing del corretto funzionamento del dominio appena discusso è stato effettuato utilizzando le istanze dei problemi definite dal secondo punto della traccia. I solver con cui si sono effettuati i test sono stati diversi: inizialmente è stato usato il solver online fornito da **editor.planning.domains**, successivamente si è passati a quello implementato mediante la libreria **PDDL4J** e a diversi dei solver messi a disposizione dal pacchetto **planutils**.

## 3 CLASSICAL PLANNING

Il secondo punto della traccia richiede di realizzare un planner e di utilizzarlo per risolvere tre istanze di problemi di complessità crescente.

### 3.1 Modellazione delle istanze

Per prima cosa è necessario modellare le condizioni iniziali e i goal di ciascuna istanza, realizzando i file **problemX.pddl**. Tale modellazione si effettua dichiarando gli oggetti necessari (insieme ai tipi di appartenenza) e istanziando i predicati dichiarati nel file di modello con gli oggetti concreti.

#### 3.1.1 Instance 1

La prima istanza ha le seguenti condizioni iniziali:

- Sono presenti 5 box, tutti ad un “depot”.
- Tutti i content da consegnare sono presenti al depot e sono sempre sufficienti.
- Le persone a cui consegnare i beni sono 3 (p1, p2 e p3), tutte situate alla stessa location. Le richieste sono le seguenti:
  - P1 ha bisogno di food e drugs.
  - P2 ha bisogno solo di drugs.
  - P3 ha bisogno solo di food.
- Sono presenti un solo robot e un solo carrier, entrambi localizzati al depot.
- La capacità del carrier è di 4 box.

Il goal dell’istanza viene raggiunto quando ad ogni persona sono stati consegnati tutti i beni di cui ha bisogno.

Il file prodotto implementando questa istanza è il seguente:

Listing 3: **problem1.pddl**

```
(define
  (problem deliveringcontents)
  (:domain emergencyServicesLogistics)

  (:objects
    depot loc1 – location
    b1 b2 b3 b4 b5 – box
    p1 p2 p3 – person
    r – robot
    car – carrier
    pl1 pl2 pl3 pl4 – place

    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 loc1)
(at p2 loc1)
(at p3 loc1)

(person-needs-content p1 food)
(person-needs-content p1 drugs)

(person-needs-content p2 drugs)

(person-needs-content p3 food)

(box-empty b1)
(box-empty b2)
(box-empty b3)
(box-empty b4)
(box-empty b5)

(place-available pl1)
(place-available pl2)
(place-available pl3)
(place-available pl4)

(place-of-carrier pl1 car)
(place-of-carrier pl2 car)
(place-of-carrier pl3 car)
(place-of-carrier pl4 car)

)

(:goal
  (and
    (person-has-content p1 food)
    (person-has-content p1 drugs)

    (person-has-content p2 drugs)

    (person-has-content p3 food)
  )
)

```

)

### 3.1.2 Instance 2

La seconda istanza si basa sulla prima, apportando solo le seguenti differenze:

- Sono presenti 2 robot e 2 carrier.
- Il numero di box presenti viene ridotto a 3.
- La capacità dei carrier viene ridotta a 2.
- Le persone a cui consegnare i beni sono 6, con le seguenti richieste:
  - P1 ha bisogno di food o tools.
  - P2 e P3 hanno bisogno di drugs.
  - P4 ha bisogno di food e drugs.
  - P5 e P6 hanno bisogno di food, drugs e tools.
- Solo P1 e P2 sono alla stessa location, le altre persone sono a location distinte.

Il goal rimane lo stesso dell'istanza precedente (ovviamente deve essere adattato alle necessità specifiche delle singole persone).

Il file prodotto implementando questa istanza è il seguente:

Listing 4: **problem2.pddl**

```
(define
  (problem deliveringcontents)
  (:domain emergencyServicesLogistics)

  (:objects
    depot loc1 loc2 loc3 loc4 loc5 – location
    b1 b2 b3 – box
    p1 p2 p3 p4 p5 p6 – person
    r1 r2 – robot
    car1 car2 – carrier
    pl1car1 pl2car1 pl1car2 pl2car2 – place

    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 loc1)
(at p2 loc1)
(at p3 loc2)
(at p4 loc3)
(at p5 loc4)
(at p6 loc5)

(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)

(box-empty b1)
(box-empty b2)
(box-empty b3)

(place-available pl1car1)
(place-available pl2car1)
(place-available pl1car2)
(place-available pl2car2)

(place-of-carrier pl1car1 car1)
(place-of-carrier pl2car1 car1)
(place-of-carrier pl1car2 car2)
(place-of-carrier pl2car2 car2)

)

(:goal

```

```

    (and
      (or (person-has-content p1 food)
          (person-has-content p1 tools))

      (person-has-content p2 drugs)

      (person-has-content p3 drugs)

      (person-has-content p4 food)
      (person-has-content p4 drugs)

      (person-has-content p5 food)
      (person-has-content p5 drugs)
      (person-has-content p5 tools)

      (person-has-content p6 food)
      (person-has-content p6 drugs)
      (person-has-content p6 tools)
    )
  )
)

```

### 3.1.3 Instance 3 (bonus)

L'istanza bonus differisce dalla seconda solo per le seguenti condizioni:

- Il numero di box viene aumentato a 4.
- Le persone a cui consegnare i beni aumentano ad 8: vengono aggiunte due persone (P7 e P8) che necessitano di food, drugs e tools.
- Solo P1 e P2 sono alla stessa location, le altre persone sono a location distinte.

Anche in questo caso il goal rimane lo stesso delle istanze precedenti.

Il file prodotto implementando questa istanza è il seguente:

Listing 5: **problem3.pddl**

```

(define
  (problem deliveringcontents)
  (:domain emergencyServicesLogistics)

  (:objects
    depot loc1 loc2 loc3 loc4 loc5 loc6 loc7 - location
    b1 b2 b3 b4 - box
    p1 p2 p3 p4 p5 p6 p7 p8 - person
    r1 r2 - robot
    car1 car2 - carrier
    pl1car1 pl2car1 pl1car2 pl2car2 - place
  )
)

```

```

    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 loc1)
  (at p2 loc1)
  (at p3 loc2)
  (at p4 loc3)
  (at p5 loc4)
  (at p6 loc5)
  (at p7 loc6)
  (at p8 loc7)

  (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)

(box-empty b1)
(box-empty b2)
(box-empty b3)
(box-empty b4)

(place-available pl1car1)
(place-available pl2car1)
(place-available pl1car2)
(place-available pl2car2)

(place-of-carrier pl1car1 car1)
(place-of-carrier pl2car1 car1)
(place-of-carrier pl1car2 car2)
(place-of-carrier pl2car2 car2)

)

(:goal
  (and
    (or (person-has-content p1 food)
        (person-has-content p1 tools))

    (person-has-content p2 drugs)

    (person-has-content p3 drugs)

    (person-has-content p4 food)
    (person-has-content p4 drugs)

    (person-has-content p5 food)
    (person-has-content p5 drugs)
    (person-has-content p5 tools)

    (person-has-content p6 food)
    (person-has-content p6 drugs)
    (person-has-content p6 tools)

    (person-has-content p7 food)
    (person-has-content p7 drugs)
    (person-has-content p7 tools)
  )
)

```

```

        (person-has-content p8 food)
        (person-has-content p8 drugs)
        (person-has-content p8 tools)
    )
)
)

```

## 3.2 Planner Custom

Il planner custom è stato realizzato usando la libreria **PDDL4J**. In particolare ne sono state realizzate due versioni, una configurata per utilizzare uno degli algoritmi di ricerca pre-implementati dalla libreria, ed una in cui l'algoritmo di ricerca è stato implementato direttamente.

Entrambe soluzioni fanno uso dell'algoritmo **A\*** e dell'euristica **Fast Forward**. Tale euristica è usata in numerosi planner e si basa sul calcolo delle soluzioni di problemi rilassati ottenuti ignorando le "delete list" degli operatori del problema originale.

### 3.2.1 Planner con algoritmo A\* di PDDL4J

Si riporta di seguito il codice sorgente del planner configurato con l'algoritmo A\* di PDDL4J.

Listing 6: **AdventurePlanner**

```

import fr.uga.pddl4j.heuristics.state.StateHeuristic;
import fr.uga.pddl4j.parser.DefaultParsedProblem;
import fr.uga.pddl4j.parser.RequireKey;
import fr.uga.pddl4j.plan.Plan;
import fr.uga.pddl4j.planners.AbstractPlanner;
import fr.uga.pddl4j.planners.SearchStrategy;
import fr.uga.pddl4j.planners.statespace.search.StateSpaceSearch;
import fr.uga.pddl4j.problem.DefaultProblem;
import fr.uga.pddl4j.problem.Problem;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import picocli.CommandLine;

@CommandLine.Command(name = "AdventurePlanner",
    version = "AdventurePlanner_1.0",
    description = "We_are_going_on_an_adventure!",
    sortOptions = false,
    mixinStandardHelpOptions = true,
    headerHeading = "Usage:%n",
    synopsisHeading = "%n",
    descriptionHeading = "%nDescription:%n%n",
    parameterListHeading = "%nParameters:%n",

```

```

        optionListHeading = "%nOptions:%n")
public class AdventurePlanner extends AbstractPlanner {

    private static final Logger LOGGER = LogManager.getLogger(
        AdventurePlanner.class.getName());

    private StateHeuristic.Name heuristic;
    private double heuristicWeight;

    public void setHeuristic(StateHeuristic.Name heuristic) {
        this.heuristic = heuristic;
    }

    public void setHeuristicWeight(final double weight) {
        if (weight <= 0) {
            throw new IllegalArgumentException("Weight_<=0");
        }
        this.heuristicWeight = weight;
    }

    public final StateHeuristic.Name getHeuristic() {
        return this.heuristic;
    }

    public final double getHeuristicWeight() {
        return this.heuristicWeight;
    }

    // Instantiates the planning problem from a parsed problem.
    @Override
    public Problem instantiate(DefaultParsedProblem problem) {
        final Problem pb = new DefaultProblem(problem);
        pb.instantiate();
        return pb;
    }

    // Solve the problem using the A* search
    @Override
    public Plan solve(final Problem problem) {
        // Creates the ASTAR search strategy
        StateSpaceSearch search = StateSpaceSearch.getInstance(
            SearchStrategy.Name.ASTAR,
            this.getHeuristic(), this.getHeuristicWeight(),
            this.getTimeout());

        LOGGER.info("*_Leaving_the_Shire_\n");
        // Search a solution

```



```

Plan plan = search.searchPlan(problem);

if (plan != null) {
    LOGGER.info ("*_Found_the_Arkenstone:_")\n");

    // Updates the planner stats
    this.getStatistics().setTimeToSearch(search.
        getSearchingTime());
    this.getStatistics().setMemoryUsedToSearch(search.
        getMemoryUsed());

    // Log the number of nodes
    LOGGER.info ("*_Created_nodes:_"+search.getCreatedNodes
        () + "\n");
    LOGGER.info ("*_Explored_nodes:_"+search.
        getExploredNodes() + "\n");
} else {
    LOGGER.info ("*_Smaug_ate_you:_")\n");
}

return plan;
}

@Override
public boolean isSupported(Problem problem) {
    return !problem.getRequirements().contains(RequireKey.
        ACTION_COSTS)
        && !problem.getRequirements().contains(RequireKey.
            CONSTRAINTS)
        && !problem.getRequirements().contains(RequireKey.
            CONTINUOUS_EFFECTS)
        && !problem.getRequirements().contains(RequireKey.
            DERIVED_PREDICATES)
        && !problem.getRequirements().contains(RequireKey.
            DURATIVE_ACTIONS)
        && !problem.getRequirements().contains(RequireKey.
            DURATION_INEQUALITIES)
        && !problem.getRequirements().contains(RequireKey.
            FLUENTS)
        && !problem.getRequirements().contains(RequireKey.
            GOAL_UTILITIES)
        && !problem.getRequirements().contains(RequireKey.
            METHOD_CONSTRAINTS)
        && !problem.getRequirements().contains(RequireKey.
            NUMERIC_FLUENTS)
        && !problem.getRequirements().contains(RequireKey.
            OBJECT_FLUENTS)

```

```

        && !problem.getRequirements().contains(RequireKey.
            PREFERENCES)
        && !problem.getRequirements().contains(RequireKey.
            TIMED_INITIAL_LITERALS)
        && !problem.getRequirements().contains(RequireKey.
            HIERARCHY);
    }

    public static void main(String[] args) {
        try {
            final AdventurePlanner planner = new AdventurePlanner
                ();

            planner.setHeuristic(StateHeuristic.Name.FAST_FORWARD)
                ;
            planner.setHeuristicWeight(1);

            CommandLine cmd = new CommandLine(planner);
            cmd.execute(args);
        } catch (IllegalArgumentException e) {
            LOGGER.fatal(e.getMessage());
        }
    }
}

```

### 3.2.2 Planner con implementazione custom di A\*

Per implementare una versione custom dell'algoritmo A\* si è reso necessario introdurre una classe di supporto **Node**, in maniera simile a quanto fatto dalla libreria.

Listing 7: **AdventurePlanner**

```

import fr.uga.pddl4j.problem.State;

public final class Node extends State
{
    private Node parent;

    // Action applied to reach this node
    private int action;

    // Cost to reach this node from the root
    private double cost;

    // Estimated distance from the goal
    private double heuristic;
}

```

```

private int depth;

public Node(State state)
{
    super(state);
}

public Node(State state, Node parent, int action, double cost,
            double heuristic)
{
    super(state);
    this.parent = parent;
    this.action = action;
    this.cost = cost;
    this.heuristic = heuristic;
    this.depth = -1;
}

public Node(State state, Node parent, int action, double cost,
            int depth, double heuristic)
{
    super(state);
    this.parent = parent;
    this.action = action;
    this.cost = cost;
    this.depth = depth;
    this.heuristic = heuristic;
}

public final int getAction()
{
    return this.action;
}

public final void setAction(final int action)
{
    this.action = action;
}

public final Node getParent()
{
    return parent;
}

public final void setParent(Node parent)
{
    this.parent = parent;
}

```

```

    }

    public final double getCost()
    {
        return cost;
    }

    public final void setCost(double cost)
    {
        this.cost = cost;
    }

    public final double getHeuristic()
    {
        return heuristic;
    }

    public final void setHeuristic(double estimates)
    {
        this.heuristic = estimates;
    }

    public int getDepth()
    {
        return this.depth;
    }

    public void setDepth(final int depth)
    {
        this.depth = depth;
    }

    public final double getValueF(double weight)
    {
        return weight * this.heuristic + this.cost;
    }
}

```

Le modifiche apportate al planner di 3.2.1 consistono nell'introduzione delle funzioni **astar** e **extractPlan**, nonché nella modifica della funzione **solve** affinché faccia uso del nuovo algoritmo di ricerca. Si riporta il codice sorgente relativo modifiche effettuate.

Listing 8: **CustomAdventurePlanner** - modifiche rispetto ad AdventurePlanner

```

// Solve the problem using the A* search
@Override
public Plan solve(final Problem problem) throws
    ProblemNotSupportedException

```

```

{
    LOGGER.info ("*_Leaving_the_Shire_\n");

    final long begin = System.currentTimeMillis();
    final Plan plan = this.astar(problem);
    final long end = System.currentTimeMillis();

    if (plan != null)
    {
        LOGGER.info ("*_Found_the_Arkenstone:_)\n");
        // Updates the planner stats
        this.getStatistics().setTimeToSearch(end - begin);
    }
    else
    {
        LOGGER.info ("*_Smaug_ate_you:_)\n");
    }

    return plan;
}

// Search a plan using A* search
public Plan astar(Problem problem) throws
    ProblemNotSupportedException
{
    if (!this.isSupported(problem))
    {
        throw new ProblemNotSupportedException ("Problem_not_
            supported");
    }

    // Create heuristic instance
    final StateHeuristic heuristic = StateHeuristic.
        getInstance(this.getHeuristic(), problem);

    // Get the initial state
    final State init = new State(problem.getInitialState());

    // Initialize the closed list of nodes (store the nodes
        explored)
    final Set<Node> close = new HashSet<>();

    // Initialize the opened list to store the pending node
    final double weight = this.getHeuristicWeight();
    final PriorityQueue<Node> open = new PriorityQueue<>(100,
        new Comparator<Node>()
    {

```

```

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);
}
});

// Create the root node and add it to the opened list
final Node root = new Node(init, null, -1, 0, heuristic.
    estimate(init, problem.getGoal()));
open.add(root);
Plan plan = null;

final int timeout = this.getTimeout() * 1000;
long time = 0;

while (!open.isEmpty() && plan == null && time < timeout)
{

    // Pop the first node in the opened list
    final Node current = open.poll();
    close.add(current);

    // If the goal is satisfied in the current node
    extract the plan from it
    if (current.satisfy(problem.getGoal()))
    {
        return this.extractPlan(current, problem);
    }
    else
    { // Else apply the actions of the problem to the
      current node
      for (int i = 0; i < problem.getActions().size(); i
          ++)
      {
          // Get the actions
          Action a = problem.getActions().get(i);
          // Apply the action if applicable
          if (a.isApplicable(current))
          {
              Node next = new Node(current);
              final List<ConditionalEffect> effects = a.
                  getConditionalEffects();
              for (ConditionalEffect ce : effects)

```

```

        {
            if (current.satisfy(ce.getCondition()))
            {
                next.apply(ce.getEffect());
            }
        }
        // Set the new node information
        final double g = current.getCost() + 1;
        if (!close.contains(next))
        {
            next.setCost(g);
            next.setParent(current);
            next.setAction(i);
            next.setHeuristic(heuristic.estimate(
                next, problem.getGoal()));
            open.add(next);
        }
    }
}

return plan;
}

// Extract the plan from the goal node following the path from
// goal to root node
private Plan extractPlan(final Node node, final Problem
    problem)
{
    Node n = node;
    final Plan plan = new SequentialPlan();
    while (n.getAction() != -1)
    {
        final Action a = problem.getActions().get(n.getAction());
        plan.add(0, a);
        n = n.getParent();
    }
    return plan;
}

```

Le due versioni si comportano in maniera molto simile, pertanto nell'analisi delle prestazioni saranno riportati i risultati relativi a una sola versione.

## 4 RISULTATI DELLE ESECUZIONI ED ANALISI DELLE PRESTAZIONI

Le istanze realizzate sono state fornite in input al solver discusso al punto 3.2 ed a diversi dei solver disponibili al pubblico, sia online che in locale sulla macchina virtuale, al fine di confrontarne i risultati e le prestazioni.

Sono stati utilizzati sia planner compatibili con **PDDL 1.2** che planner compatibili esclusivamente con le **durative-action** introdotte dalla versione **2.1**. Per effettuare i test su quest'ultimo tipo di planner si è reso necessario convertire il dominio descritto precedentemente in uno che facesse uso delle **durative-action**.

### 4.1 Conversione del dominio in PDDL 2.1

La conversione è stata effettuata apportando le seguenti modifiche:

- Tutte le azioni “classiche” sono state trasformate in **durative-action**, utilizzando delle durate “fake”. Si è scelto di fissare durate unitarie per tutte le operazioni di carico/scarico di box e content, mentre sono state utilizzate durate maggiori (comunque fisse) per gli spostamenti.
- Condizioni ed effetti ora fanno uso degli indicatori temporali **at start**, **at end** e **over all**, che indicano che una data condizione deve essere verificata (o viene applicata) all’inizio dell’azione, alla fine dell’azione o durante tutta l’azione. Tali indicatori sono stati inseriti in maniera opportuna alle condizioni ed agli effetti già presenti nel modello 1.2.
- Alla luce delle limitazioni relative al parsing delle **negative-precondition** (utilizzate nelle azioni di **load-one-box-on-carrier** e **unload-one-box-from-carrier**) di alcuni dei solver da utilizzare, si è introdotto un predicato **place-occupied** complementare a **place-available** e le azioni sono state modificate di conseguenza.
- Dato che le azioni non sono più considerabili istantanee come in PDDL 1.2 è stato necessario introdurre un nuovo predicato **robot-free** in grado di modellare quando un dato agente robotico è libero e quindi in grado di eseguire una nuova azione.
- PDDL 2.1 abilita inoltre all’uso dei **fluents**, pertanto si è deciso di mantenere un fluent **path-cost** che indica il costo accumulato da ogni robot, equivalente alla durata delle azioni effettuate. Nei file di dominio è possibile utilizzare tale fluent per specificare una **metrica** che indica, ai soli solver in grado di ottimizzare la soluzione trovata, il valore in base al quale provare ad ottimizzare. Nel caso di un solo agente robotico l’istruzione utilizzata è:

(**:metric minimize** (path-cost r))

Mentre nel caso di più agenti è possibile utilizzare:

(**:metric minimize** (+ (path-cost r1) (path-cost r2)))

Il dominio convertito ottenuto è il seguente:

Listing 9: **converted-domain.pddl**

( <b>define</b> ( <b>domain</b> emergencyServicesLogistics)
---



```

(:requirements
  :adl
  :universal-preconditions
  :durative-actions
  :numeric-fluents
  :duration-inequalities
)

(:types
  location locatable place - object
  moving not-moving - locatable
  person - not-moving
  box content robot carrier - moving
)

(:functions
  (path-cost ?r - robot)
)

(:predicates
  (at ?obj - locatable ?loc - location)

  (box-empty ?box - box)
  (box-contains ?box - box ?c - content )

  (box-loaded-on-carrier ?box - box ?car - carrier)

  (person-has-content ?p - person ?c - content)
  (person-needs-content ?p - person ?c - content)

  (place-available ?pl - place)
  (place-occupied ?pl - place)
  (place-of-carrier ?pl - place ?car - carrier)

  (robot-free ?r - robot)
)

(:durative-action fill-box-with-content
  :parameters
    (?b - box
      ?c - content
      ?r - robot
      ?loc -location
    )
)

```

```

:duration(= ?duration 1)

:condition
  (and
    (at start (box-empty ?b))
    (over all (at ?c ?loc))
    (over all (at ?b ?loc))
    (over all (at ?r ?loc))
    (at start (robot-free ?r))
  )
:effect
  (and
    (at start (not (robot-free ?r)))
    (at end (robot-free ?r))
    (at end (box-contains ?b ?c))
    (at start (not (box-empty ?b)))

    (at end (increase (path-cost ?r) 1))
  )
)

(:durative-action empty-box-leaving-content
:parameters
  (?b - box
   ?c - content
   ?r - robot
   ?p - person
   ?loc -location
  )

:duration(= ?duration 1)

:condition
  (and
    (at start (box-contains ?b ?c))
    (over all (at ?b ?loc))
    (over all (at ?r ?loc))
    (over all (at ?p ?loc))
    (over all (person-needs-content ?p ?c))
    (at start (robot-free ?r))
  )
:effect
  (and
    (at start (not (robot-free ?r)))
    (at end (robot-free ?r))
    (at start (not (box-contains ?b ?c)))
  )
)

```

```

        (at end (box-empty ?b))
        (at end (person-has-content ?p ?c))

        (at end (increase (path-cost ?r) 1))
    )
)

(:durative-action load-one-box-on-carrier
  :parameters
    (?b - box
     ?car - carrier
     ?r - robot
     ?loc -location
     ?pl - place
    )

  :duration(= ?duration 1)

  :condition
    (and
      (at start (at ?b ?loc))
      (over all (at ?car ?loc))
      (over all (at ?r ?loc))
      (over all (place-of-carrier ?pl ?car))
      (at start (place-available ?pl))
      (at start (robot-free ?r))
    )

  :effect
    (and
      (at start (not (robot-free ?r)))
      (at end (robot-free ?r))
      (at start (not (at ?b ?loc)))
      (at end (box-loaded-on-carrier ?b ?car))
      (at start (not (place-available ?pl)))
      (at start (place-occupied ?pl))

      (at end (increase (path-cost ?r) 1))
    )
)

(:durative-action move-robot
  :parameters
    (?r - robot
     ?from - location
     ?to - location
    )
)

```

```

:duration(= ?duration 8)

:condition
  (and
    (at start (at ?r ?from))
    (at start (robot-free ?r))
  )

:effect
  (and
    (at start (not (robot-free ?r)))
    (at end (robot-free ?r))
    (at start (not (at ?r ?from)))
    (at end (at ?r ?to))

    (at end (increase (path-cost ?r) 8))
  )
)

(:durative-action move-robot-carrier
  :parameters
    (?r - robot
     ?car - carrier
     ?from - location
     ?to - location
    )

  :duration(= ?duration 10)

  :condition
    (and
      (at start (at ?r ?from))
      (at start (at ?car ?from))
      (at start (robot-free ?r))
    )

  :effect
    (and
      (at start (not (robot-free ?r)))
      (at end (robot-free ?r))
      (at start (not (at ?r ?from)))
      (at end (at ?r ?to))
      (at start (not (at ?car ?from)))
      (at end (at ?car ?to))

      (at end (increase (path-cost ?r) 10))
    )
)

```

```

    )
)

(:durative-action unload-one-box-from-carrier
 :parameters
  (?b - box
   ?car - carrier
   ?r - robot
   ?loc -location
   ?pl - place
  )

 :duration(= ?duration 1)

 :condition
  (and
    (over all (at ?car ?loc))
    (over all (at ?r ?loc))
    (at start (box-loaded-on-carrier ?b ?car))
    (over all (place-of-carrier ?pl ?car))
    (over all (place-occupied ?pl))
    (at start (robot-free ?r))
  )

 :effect
  (and
    (at start (not (robot-free ?r)))
    (at end (robot-free ?r))
    (at end (at ?b ?loc))
    (at start (not (box-loaded-on-carrier ?b ?car)))
    (at end (not (place-occupied ?pl)))
    (at end (place-available ?pl))

    (at end (increase (path-cost ?r) 1))
  )
)
)

```

I file di problema hanno subito leggere modifiche per garantire la compatibilità con il nuovo dominio.

## 4.2 Risultati e limitazioni dei solver utilizzati

È risultato subito evidente come tutti i solver eseguiti in locale siano soggetti a pesanti limitazioni a causa delle ridotte risorse hardware a disposizione delle macchine utilizzate (nonostante gli 8GB di RAM allocati alla macchina virtuale) e che, per ottenere risultati significativi per le istanze di dimensione maggiore, sarebbe stato necessario utilizzare macchine notevolmente più prestanti in termini di hardware (almeno 32GB

di RAM). Ovvvia eccezione è quella del solver online, che esegue su server ad elevate prestazioni, tuttavia soggetto a limitazioni temporali.

#### 4.2.1 OR nei goal delle istanze 2 e 3

È importante segnalare come persino il solver online, il più prestante di quelli a disposizione, non sia stato in grado di portare a termine la ricerca sulle istanze 2 e 3 in presenza dell'**or** nei goal. Pertanto si è deciso, al fine di alleggerire il carico di lavoro dei solver ed ottenere dei risultati da poter analizzare, di effettuare una prima semplificazione delle istanze rimuovendo gli **or** relativi a P1: nelle istanze modificate P1 necessiterà **sia di food che di tools**.

#### 4.2.2 Solver utilizzati e anteprima dei risultati

Sono stati utilizzati svariati solver, sia per la versione 1.2 che per la 2.1. Alcuni di essi tentano di trovare soluzioni ottime, mentre altri si limitano a soluzioni in grado di soddisfare i goal, che spesso risultano essere ben distanti dalle versioni desiderate.

Seguono delle tabelle riassuntive dei risultati ottenuti. Per evitare che i confronti tra i risultati siano influenzati dalla versione di PDDL utilizzata, si è deciso di suddividere i solver in base a tale parametro.

Ciascuno dei solver utilizzati e le eventuali soluzioni trovate saranno discusse successivamente.

Solver	Ottimizza	Istanza 1	Plan ottimale	Istanza 2	Plan ottimale
Solver online	No	< 1 sec	No	≈ 1 sec	No
Solver custom	Sì	≈ 40 sec	Sì	Memoria <sup>1</sup>	–
Solver custom w <sup>2</sup>	No	1 - 18 sec	No	≈ 7 sec	No
Delfi	Sì	≈ 1 sec	Sì	Memoria	–
Scorpion	Sì	≈ 5 min	Sì	–	–
Cerberus	No	≈ 2 min	No	–	–

Tabella 4.1: Solver PDDL 1.2 (istanze 1 e 2)

Solver	Ottimizza	Istanza 3	Plan ottimale
Solver online	No	≈ 5 sec	No
Solver custom	Sì	–	–
Solver custom w	No	No solution found	–
Delfi	Sì	–	–
Scorpion	Sì	–	–
Cerberus	No	–	–

Tabella 4.2: Solver PDDL 1.2 (istanza 3)

<sup>1</sup>Il planner termina la RAM a disposizione.

<sup>2</sup>Esecuzioni con peso dell'euristica non unitario

Solver	Ottimizza	Istanza 1	Plan ottimale	Istanza 2	Plan ottimale
Popf	No	< 1 sec	No	Memoria	–
Optic	No	Memoria	–	–	–
Lpg-td -speed	No	< 1 sec	No	≈ 35 min	No
Lpg-td -quality	Sì	≈ 5 sec	Sì	–	–

Tabella 4.3: Solver PDDL 2.1 con durative-actions (istanze 1 e 2)

#### 4.2.3 Solver online: *solver.planning.domains*

Il solver online testato è **solver.planning.domains**, utilizzato dall'editor di testo disponibile a **editor.planning.domains**. Riesce a risolvere tutte e 3 le istanze proposte in quanto non soggetto alle limitazioni di risorse precedentemente discusse. Tuttavia, usando un algoritmo di tipo **BFS**, i plan restituiti da questo solver sono ben lontani da quelli desiderabili (ne sono un esempio le move multiple nella soluzione della prima istanza, in cui ne basterebbe una sola).

Per brevità di trattazione si riporta solo il primo dei plan generati.

Listing 10: **Solver online** - Plan problem 1

```
(load-one-box-on-carrier b3 car r depot pl1)
(fill-box-with-content b5 food r depot)
(load-one-box-on-carrier b5 car r depot pl4)
(move-robot-carrier r car depot loc1)
(unload-one-box-from-carrier b5 car r loc1 pl1)
(move-robot-carrier r car loc1 depot)
(fill-box-with-content b4 drugs r depot)
(load-one-box-on-carrier b4 car r depot pl3)
(move-robot-carrier r car depot loc1)
(empty-box-leaving-content b5 food r pl loc1)
(unload-one-box-from-carrier b4 car r loc1 pl3)
(empty-box-leaving-content b4 drugs r pl loc1)
(move-robot-carrier r car loc1 depot)
(fill-box-with-content b1 food r depot)
(load-one-box-on-carrier b1 car r depot pl3)
(move-robot-carrier r car depot loc1)
(unload-one-box-from-carrier b1 car r loc1 pl3)
(empty-box-leaving-content b1 food r p3 loc1)
(move-robot-carrier r car loc1 depot)
(fill-box-with-content b2 drugs r depot)
(load-one-box-on-carrier b2 car r depot pl3)
(move-robot-carrier r car depot loc1)
(unload-one-box-from-carrier b2 car r loc1 pl3)
(empty-box-leaving-content b2 drugs r p2 loc1)
```

#### 4.2.4 Planner custom

Il planner custom descritto in 3.2 è stato eseguito inizialmente con **peso dell'euristica pari a 1**. Così facendo è stato in grado di fornire un plan **ottimale** per l'istanza 1 in circa 40 secondi. Si riporta il plan ottimale ottenuto.

Listing 11: **Solver custom con peso dell'euristica unitario** - Plan problem 1

```
problem instantiation done successfully (123 actions, 61 fluents)
```

```
* Leaving the Shire
* Found the Arkenstone :)
* Created nodes: 12233629
* Explored nodes: 1106764
```

found plan as follows:

```
00: (      fill-box-with-content b3 food r depot) [0]
01: (      fill-box-with-content b1 drugs r depot) [0]
02: (  load-one-box-on-carrier b3 car r depot pl1) [0]
03: (  load-one-box-on-carrier b1 car r depot pl3) [0]
04: (      fill-box-with-content b2 drugs r depot) [0]
05: (      fill-box-with-content b4 food r depot) [0]
06: (  load-one-box-on-carrier b4 car r depot pl4) [0]
07: (  load-one-box-on-carrier b2 car r depot pl2) [0]
08: (      move-robot-carrier r car depot loc1) [0]
09: (unload-one-box-from-carrier b1 car r loc1 pl1) [0]
10: (unload-one-box-from-carrier b3 car r loc1 pl3) [0]
11: (unload-one-box-from-carrier b4 car r loc1 pl4) [0]
12: (  empty-box-leaving-content b3 food r p1 loc1) [0]
13: (  empty-box-leaving-content b4 food r p3 loc1) [0]
14: (  empty-box-leaving-content b1 drugs r p1 loc1) [0]
15: (unload-one-box-from-carrier b2 car r loc1 pl2) [0]
16: (  empty-box-leaving-content b2 drugs r p2 loc1) [0]
```

```
time spent:      0,05 seconds parsing
                0,09 seconds encoding
                40,68 seconds searching
                40,82 seconds total time
```

```
memory used:     0,53 MBytes for problem representation
                428,83 MBytes for searching
                429,36 MBytes total
```

Tuttavia durante l'esecuzione sull'istanza 2 è stata terminata la memoria a disposizione.

Si è quindi deciso di eseguire il medesimo planner ma **aumentando il peso dato all'euristica**. Assegnando un peso maggiore di 1 si accettano **soluzioni sub-ottime**, ma si registrano notevoli **miglioramenti in termini di tempi di esecuzione e memoria oc-**



**cupata.** Testando questo approccio sull'istanza 1 si sono ottenuti tempi di esecuzione variabili, dai 18 secondi ottenuti fissando il peso a 1.5, al secondo ottenuto con peso 4. È tuttavia doveroso segnalare come all'aumentare del peso dell'euristica la qualità dei plan restituiti vada a diminuire, pertanto è necessario trovare il giusto compromesso tra qualità del plan e prestazioni del planner. Si riportano a titolo esemplificativo due dei plan generati.

Listing 12: **Solver custom con peso dell'euristica 1.5** - Plan problem 1

```

problem instantiation done successfully (123 actions, 61 fluents)

* Leaving the Shire
* Found the Arkenstone :)
* Created nodes: 5510362
* Explored nodes: 475214

found plan as follows:

00: (      fill-box-with-content b1 food r depot) [0]
01: (      fill-box-with-content b4 drugs r depot) [0]
02: (  load-one-box-on-carrier b4 car r depot pl2) [0]
03: (  load-one-box-on-carrier b1 car r depot pl4) [0]
04: (      fill-box-with-content b5 food r depot) [0]
05: (  load-one-box-on-carrier b5 car r depot pl1) [0]
06: (      move-robot-carrier r car depot loc1) [0]
07: (unload-one-box-from-carrier b5 car r loc1 pl4) [0]
08: (unload-one-box-from-carrier b4 car r loc1 pl2) [0]
09: (  empty-box-leaving-content b5 food r p3 loc1) [0]
10: (unload-one-box-from-carrier b1 car r loc1 pl1) [0]
11: (  empty-box-leaving-content b1 food r p1 loc1) [0]
12: (      move-robot-carrier r car loc1 depot) [0]
13: (      fill-box-with-content b2 drugs r depot) [0]
14: (  load-one-box-on-carrier b2 car r depot pl1) [0]
15: (      move-robot-carrier r car depot loc1) [0]
16: (unload-one-box-from-carrier b2 car r loc1 pl1) [0]
17: (  empty-box-leaving-content b2 drugs r p1 loc1) [0]
18: (  empty-box-leaving-content b4 drugs r p2 loc1) [0]

time spent:      0,05 seconds parsing
                0,08 seconds encoding
                18,21 seconds searching
                18,34 seconds total time

memory used:     0,53 MBytes for problem representation
                205,71 MBytes for searching
                206,24 MBytes total

```

Listing 13: **Solver custom con peso dell'euristica 4** - Plan problem 1

**problem** instantiation done successfully (123 actions, 61 fluents)

- \* Leaving the Shire
- \* Found the Arkenstone :)
- \* Created nodes: 84469
- \* Explored nodes: 7449

found plan as follows:

```
00: (      fill-box-with-content b3 food r depot) [0]
01: (      fill-box-with-content b1 drugs r depot) [0]
02: (  load-one-box-on-carrier b1 car r depot pl1) [0]
03: (  load-one-box-on-carrier b3 car r depot pl2) [0]
04: (      move-robot-carrier r car depot loc1) [0]
05: (unload-one-box-from-carrier b1 car r loc1 pl1) [0]
06: (unload-one-box-from-carrier b3 car r loc1 pl2) [0]
07: (      move-robot-carrier r car loc1 depot) [0]
08: (      fill-box-with-content b2 food r depot) [0]
09: (  load-one-box-on-carrier b2 car r depot pl3) [0]
10: (      move-robot-carrier r car depot loc1) [0]
11: ( empty-box-leaving-content b3 food r pl loc1) [0]
12: (unload-one-box-from-carrier b2 car r loc1 pl3) [0]
13: ( empty-box-leaving-content b2 food r p3 loc1) [0]
14: (      move-robot-carrier r car loc1 depot) [0]
15: (      fill-box-with-content b5 drugs r depot) [0]
16: (  load-one-box-on-carrier b5 car r depot pl1) [0]
17: (      move-robot-carrier r car depot loc1) [0]
18: (unload-one-box-from-carrier b5 car r loc1 pl1) [0]
19: ( empty-box-leaving-content b1 drugs r pl loc1) [0]
20: ( empty-box-leaving-content b5 drugs r p2 loc1) [0]
```

time spent:            0,04 seconds parsing  
                      0,08 seconds encoding  
                      0,78 seconds searching  
                      0,90 seconds total time

memory used:           0,53 MBytes for **problem** representation  
                      3,46 MBytes for searching  
                      3,99 MBytes total

Per quanto riguarda l'istanza 2, si è reso necessario aumentare ulteriormente il peso assegnato all'euristica, arrivando a 10. Così facendo si è ottenuta una soluzione in circa 7 secondi, tuttavia tale soluzione è ben distante dall'essere ottima, anche a causa della presenza di diversi spostamenti superflui. Si riporta il plan generato.

Listing 14: **Solver custom con peso dell'euristica 10** - Plan problem 2

**problem** instantiation done successfully (594 actions, 101 fluents)

- \* Leaving the Shire
- \* Found the Arkenstone :)
- \* Created nodes: 652969
- \* Explored nodes: 33649

found plan as follows:

```

000: (                                move-robot r1 depot loc4) [0]
001: (    load-one-box-on-carrier b2 car2 r2 depot pl1car2) [0]
002: (    load-one-box-on-carrier b1 car2 r2 depot pl2car2) [0]
003: (                                move-robot-carrier r2 car2 depot loc2) [0]
004: (                                move-robot r2 loc2 depot) [0]
005: (                                fill-box-with-content b3 drugs r2 depot) [0]
006: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
007: (                                move-robot-carrier r2 car1 depot loc3) [0]
008: ( unload-one-box-from-carrier b3 car1 r2 loc3 pl1car1) [0]
009: (    empty-box-leaving-content b3 drugs r2 p4 loc3) [0]
010: (    load-one-box-on-carrier b3 car1 r2 loc3 pl2car1) [0]
011: (                                move-robot-carrier r2 car1 loc3 depot) [0]
012: (unload-one-box-from-carrier b3 car1 r2 depot pl2car1) [0]
013: (                                fill-box-with-content b3 drugs r2 depot) [0]
014: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
015: (                                move-robot-carrier r2 car1 depot loc5) [0]
016: ( unload-one-box-from-carrier b3 car1 r2 loc5 pl1car1) [0]
017: (    empty-box-leaving-content b3 drugs r2 p6 loc5) [0]
018: (    load-one-box-on-carrier b3 car1 r2 loc5 pl2car1) [0]
019: (                                move-robot-carrier r2 car1 loc5 depot) [0]
020: (unload-one-box-from-carrier b3 car1 r2 depot pl2car1) [0]
021: (                                fill-box-with-content b3 drugs r2 depot) [0]
022: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
023: (                                move-robot-carrier r2 car1 depot loc1) [0]
024: ( unload-one-box-from-carrier b3 car1 r2 loc1 pl1car1) [0]
025: (    empty-box-leaving-content b3 drugs r2 p2 loc1) [0]
026: (    load-one-box-on-carrier b3 car1 r2 loc1 pl2car1) [0]
027: (                                move-robot-carrier r2 car1 loc1 depot) [0]
028: (unload-one-box-from-carrier b3 car1 r2 depot pl2car1) [0]
029: (                                fill-box-with-content b3 drugs r2 depot) [0]
030: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
031: (                                move-robot-carrier r2 car1 depot loc3) [0]
032: (                                move-robot r1 loc4 loc1) [0]
033: (                                move-robot-carrier r2 car1 loc3 loc4) [0]
034: ( unload-one-box-from-carrier b3 car1 r2 loc4 pl1car1) [0]
035: (    empty-box-leaving-content b3 drugs r2 p5 loc4) [0]
036: (                                move-robot r2 loc4 loc2) [0]
037: (                                move-robot-carrier r2 car2 loc2 depot) [0]
038: (unload-one-box-from-carrier b2 car2 r2 depot pl2car2) [0]
039: (                                fill-box-with-content b2 drugs r2 depot) [0]

```

```

040: (    load-one-box-on-carrier b2 car2 r2 depot pl2car2) [0]
041: (          move-robot-carrier r2 car2 depot loc2) [0]
042: ( unload-one-box-from-carrier b2 car2 r2 loc2 pl1car2) [0]
043: (          empty-box-leaving-content b2 drugs r2 p3 loc2) [0]
044: (          move-robot r2 loc2 loc4) [0]
045: (    load-one-box-on-carrier b3 car1 r2 loc4 pl1car1) [0]
046: (          move-robot-carrier r2 car1 loc4 depot) [0]
047: (unload-one-box-from-carrier b3 car1 r2 depot pl1car1) [0]
048: (          fill-box-with-content b3 food r2 depot) [0]
049: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
050: (          move-robot-carrier r2 car1 depot loc3) [0]
051: ( unload-one-box-from-carrier b3 car1 r2 loc3 pl1car1) [0]
052: (          empty-box-leaving-content b3 food r2 p4 loc3) [0]
053: (    load-one-box-on-carrier b3 car1 r2 loc3 pl1car1) [0]
054: (          move-robot-carrier r2 car1 loc3 depot) [0]
055: (unload-one-box-from-carrier b3 car1 r2 depot pl1car1) [0]
056: (          fill-box-with-content b3 tools r2 depot) [0]
057: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
058: (          move-robot-carrier r2 car1 depot loc4) [0]
059: ( unload-one-box-from-carrier b3 car1 r2 loc4 pl1car1) [0]
060: (          empty-box-leaving-content b3 tools r2 p5 loc4) [0]
061: (    load-one-box-on-carrier b3 car1 r2 loc4 pl1car1) [0]
062: (          move-robot-carrier r2 car1 loc4 depot) [0]
063: (unload-one-box-from-carrier b3 car1 r2 depot pl1car1) [0]
064: (          fill-box-with-content b3 food r2 depot) [0]
065: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
066: (          move-robot-carrier r2 car1 depot loc4) [0]
067: ( unload-one-box-from-carrier b3 car1 r2 loc4 pl1car1) [0]
068: (          empty-box-leaving-content b3 food r2 p5 loc4) [0]
069: (    load-one-box-on-carrier b3 car1 r2 loc4 pl1car1) [0]
070: (          move-robot-carrier r2 car1 loc4 depot) [0]
071: (unload-one-box-from-carrier b3 car1 r2 depot pl1car1) [0]
072: (          fill-box-with-content b3 food r2 depot) [0]
073: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
074: (          move-robot-carrier r2 car1 depot loc5) [0]
075: ( unload-one-box-from-carrier b3 car1 r2 loc5 pl1car1) [0]
076: (          empty-box-leaving-content b3 food r2 p6 loc5) [0]
077: (    load-one-box-on-carrier b3 car1 r2 loc5 pl1car1) [0]
078: (          move-robot-carrier r2 car1 loc5 depot) [0]
079: (unload-one-box-from-carrier b3 car1 r2 depot pl1car1) [0]
080: (          fill-box-with-content b3 tools r2 depot) [0]
081: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
082: (          move-robot-carrier r2 car1 depot loc5) [0]
083: ( unload-one-box-from-carrier b3 car1 r2 loc5 pl1car1) [0]
084: (          empty-box-leaving-content b3 tools r2 p6 loc5) [0]
085: (    load-one-box-on-carrier b3 car1 r2 loc5 pl1car1) [0]
086: (          move-robot-carrier r2 car1 loc5 depot) [0]

```

```

087: (unload-one-box-from-carrier b3 car1 r2 depot pl1car1) [0]
088: (          fill-box-with-content b3 food r2 depot) [0]
089: (    load-one-box-on-carrier b3 car1 r2 depot pl1car1) [0]
090: (          move-robot r1 loc1 depot) [0]
091: (          move-robot-carrier r1 car1 depot loc1) [0]
092: ( unload-one-box-from-carrier b3 car1 r1 loc1 pl1car1) [0]
093: (    empty-box-leaving-content b3 food r1 p1 loc1) [0]
094: (          move-robot r2 depot loc2) [0]
095: (          move-robot-carrier r2 car2 loc2 depot) [0]
096: (unload-one-box-from-carrier b1 car2 r2 depot pl2car2) [0]
097: (          fill-box-with-content b1 tools r2 depot) [0]
098: (    load-one-box-on-carrier b1 car2 r2 depot pl1car2) [0]
099: (          move-robot-carrier r2 car2 depot loc1) [0]
100: ( unload-one-box-from-carrier b1 car2 r1 loc1 pl1car2) [0]
101: (    empty-box-leaving-content b1 tools r1 p1 loc1) [0]

time spent:      0,05 seconds parsing
                  0,18 seconds encoding
                  6,72 seconds searching
                  6,95 seconds total time

memory used:     2,00 MBytes for problem representation
                  13,63 MBytes for searching
                  15,63 MBytes total

```

Sono stati effettuati test anche con valori differenti, tuttavia con pesi pari ad 8 e 6,5 si sono ottenute soluzioni peggiori dal punto di vista del numero di azioni a fronte di un peggioramento anche in termini di tempi di esecuzione e memoria consumata. Pertanto si è ritenuto più adeguato riportare la soluzione ottenuta con peso 10. Infine, in esecuzione sull'istanza 3 il planner non è stato in grado di trovare una soluzione, probabilmente a causa del peso eccessivo assegnato all'euristica. Tuttavia, l'assegnamento di pesi minori comporterebbe l'esaurimento della memoria a disposizione, pertanto non si è prodotto alcun plan per questa istanza.

#### 4.2.5 Delfi

Delfi è uno dei solver messi a disposizione dalla libreria **planutils**. Consiste in un portfolio di diversi planner, ed utilizza tecniche di **deep learning** per creare un modello in base al quale scegliere quali planner utilizzare. È in grado di restituire **soluzioni ottimali** in termini di costo ed è più efficiente anche dei solver che non effettuano alcuna ottimizzazione. Pertanto risulta essere il **più promettente** dei solver testati su PDDL 1.2.

La prima istanza è stata risolta in tempi (reali) prossimi al secondo.

Listing 15: **Delfi** - Plan problem 1

```

Solution found!
Actual search time: 0.148846s [t=0.375128s]
fill-box-with-content b1 drugs r depot (1)

```

```
fill-box-with-content b2 food r depot (1)
load-one-box-on-carrier b2 car r depot pl1 (1)
load-one-box-on-carrier b1 car r depot pl2 (1)
fill-box-with-content b3 drugs r depot (1)
load-one-box-on-carrier b3 car r depot pl3 (1)
fill-box-with-content b4 food r depot (1)
load-one-box-on-carrier b4 car r depot pl4 (1)
move-robot-carrier r car depot loc1 (1)
unload-one-box-from-carrier b2 car r loc1 pl1 (1)
empty-box-leaving-content b2 food r pl loc1 (1)
unload-one-box-from-carrier b1 car r loc1 pl2 (1)
empty-box-leaving-content b1 drugs r pl loc1 (1)
unload-one-box-from-carrier b3 car r loc1 pl3 (1)
empty-box-leaving-content b3 drugs r p2 loc1 (1)
unload-one-box-from-carrier b4 car r loc1 pl4 (1)
empty-box-leaving-content b4 food r p3 loc1 (1)
```

Plan length: 17 step(s).

Plan cost: 17

Expanded 13910 state(s).

Reopened 0 state(s).

Evaluated 19162 state(s).

Evaluations: 19162

Generated 111969 state(s).

Dead ends: 0 state(s).

Expanded until last jump: 13890 state(s).

Reopened until last jump: 0 state(s).

Evaluated until last jump: 19107 state(s).

Generated until last jump: 111745 state(s).

Number of registered states: 19162

Search time: 0.149415s

Total time: 0.375135s

Solution found.

Peak memory: 11208 KB

Overall time: [0.520s CPU, 0.590s wall-clock]

Tuttavia, durante l'esecuzione della seconda istanza, il planner ha consumato tutta la memoria a disposizione della macchina, arrestandosi con un errore.

#### 4.2.6 *Scorpion*

Altro solver messo a disposizione dalla libreria **planutils**. Anch'esso in grado di restituire **soluzioni ottimali**, utilizza il sistema **Fast Downward**. Risulta tuttavia essere **notevolmente meno efficiente dell'equivalente Delfi** già dalla prima istanza, pertan-

to non sono stati effettuati test sulle istanze successive. Il piano restituito è del tutto analogo a quello fornito da Delfi.

#### 4.2.7 Cerberus

Solver di **planutils** che fa uso di un'euristica **red-black**. Nonostante non effettui particolari ottimizzazioni sul risultato trovato, risulta **molto meno efficiente di Delfi**, pertanto il testing si è limitato alla prima istanza. Il piano restituito non differisce in maniera significativa dai precedenti.

#### 4.2.8 Popf

È uno dei planner temporali accessibili da **planutils**. Si tratta di un "**satisficing**" **planner**, cioè di un planner che punta solo a soddisfare i goal senza cercare ottimi. Pertanto, nonostante sia stata indicata una metrica nel file di problema, il plan restituito risulta essere ben distante da uno ottimale, in maniera analoga a quanto visto per altri solver di questo tipo. I tempi di esecuzione della prima istanza sono inferiori al secondo, tuttavia eseguendo sulla seconda istanza viene terminata la memoria a disposizione.

Listing 16: **Popf** - Plan problem 1

```
0.000: (fill-box-with-content b1 food r depot) [1.000]
1.001: (load-one-box-on-carrier b1 car r depot pl1) [1.000]
2.002: (fill-box-with-content b2 drugs r depot) [1.000]
3.003: (load-one-box-on-carrier b2 car r depot pl2) [1.000]
4.004: (move-robot-carrier r car depot loc1) [10.000]
14.005: (unload-one-box-from-carrier b2 car r loc1 pl1) [1.000]
15.006: (unload-one-box-from-carrier b1 car r loc1 pl2) [1.000]
16.007: (empty-box-leaving-content b1 food r p3 loc1) [1.000]
17.008: (move-robot-carrier r car loc1 depot) [10.000]
27.009: (fill-box-with-content b3 food r depot) [1.000]
28.010: (load-one-box-on-carrier b3 car r depot pl1) [1.000]
29.011: (move-robot-carrier r car depot loc1) [10.000]
39.012: (unload-one-box-from-carrier b3 car r loc1 pl1) [1.000]
40.013: (empty-box-leaving-content b3 food r p1 loc1) [1.000]
41.014: (empty-box-leaving-content b2 drugs r p2 loc1) [1.000]
42.015: (move-robot-carrier r car loc1 depot) [10.000]
52.016: (fill-box-with-content b4 drugs r depot) [1.000]
53.017: (load-one-box-on-carrier b4 car r depot pl1) [1.000]
54.018: (move-robot-carrier r car depot loc1) [10.000]
64.019: (unload-one-box-from-carrier b4 car r loc1 pl1) [1.000]
65.020: (empty-box-leaving-content b4 drugs r p1 loc1) [1.000]
```

#### 4.2.9 Optic

Planner temporale in teoria in grado di ottimizzare la soluzione. Termina la memoria già dalla prima istanza.

#### 4.2.10 Lpg-td

Lpg-td (Local search for Planning Graphs) è un planner temporale basato su **local search** disponibile attraverso **planutils**. È possibile utilizzarlo in modalità **-speed**, nella quale si ferma alla prima soluzione disponibile, o in modalità **-quality**, nella quale trova una soluzione (come in -speed) e poi prova a migliorarla utilizzando un certo tempo di CPU.

L'esecuzione in modalità speed della prima istanza termina in meno di un secondo. Il plan restituito non risulta essere ottimale a causa della presenza di una fill superflua, tuttavia rimane molto vicino all'ottimo.

Listing 17: **Lpg-td -speed** - Plan problem 1

```
; Version LPG-td-1.4

; Time 0.31
; Search time 0.30
; Parsing time 0.01
; Mutex time 0.00
; MetricValue 27.00

0.0002: (FILL-BOX-WITH-CONTENT B2 DRUGS R DEPOT) [1.0000]
1.0005: (FILL-BOX-WITH-CONTENT B1 DRUGS R DEPOT) [1.0000]
2.0008: (FILL-BOX-WITH-CONTENT B3 FOOD R DEPOT) [1.0000]
3.0010: (FILL-BOX-WITH-CONTENT B4 FOOD R DEPOT) [1.0000]
4.0012: (FILL-BOX-WITH-CONTENT B5 DRUGS R DEPOT) [1.0000]
5.0015: (LOAD-ONE-BOX-ON-CARRIER B3 CAR R DEPOT PL1) [1.0000]
6.0017: (LOAD-ONE-BOX-ON-CARRIER B4 CAR R DEPOT PL2) [1.0000]
7.0020: (LOAD-ONE-BOX-ON-CARRIER B1 CAR R DEPOT PL3) [1.0000]
8.0022: (LOAD-ONE-BOX-ON-CARRIER B2 CAR R DEPOT PL4) [1.0000]
9.0025: (MOVE-ROBOT-CARRIER R CAR DEPOT LOC1) [10.0000]
19.0027: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR R LOC1 PL3)
        [1.0000]
20.0030: (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR R LOC1 PL1)
        [1.0000]
21.0032: (EMPTY-BOX-LEAVING-CONTENT B2 DRUGS R P2 LOC1) [1.0000]
22.0035: (EMPTY-BOX-LEAVING-CONTENT B1 DRUGS R P1 LOC1) [1.0000]
23.0037: (UNLOAD-ONE-BOX-FROM-CARRIER B4 CAR R LOC1 PL4)
        [1.0000]
24.0040: (EMPTY-BOX-LEAVING-CONTENT B4 FOOD R P1 LOC1) [1.0000]
25.0042: (UNLOAD-ONE-BOX-FROM-CARRIER B3 CAR R LOC1 PL2)
        [1.0000]
26.0045: (EMPTY-BOX-LEAVING-CONTENT B3 FOOD R P3 LOC1) [1.0000]
```

Eseguendo la prima istanza in modalità quality il risultato viene restituito in circa 5 secondi. La rimozione della fill superflua comporta il decremento della metrica (da 27 a 26) e il raggiungimento di un plan ottimo.

Listing 18: **Lpg-td -quality** - Plan problem 1

```
; Version LPG-td-1.4
```



```

; Time 4.76
; Plan generation time 3.37
; Search time 3.37
; Parsing time 0.00
; Mutex time 0.00
; MetricValue 26.00

0.0003 (FILL-BOX-WITH-CONTENT B1 DRUGS R DEPOT) [1.0000]
1.0005 (LOAD-ONE-BOX-ON-CARRIER B1 CAR R DEPOT PL3) [1.0000]
2.0008 (FILL-BOX-WITH-CONTENT B4 FOOD R DEPOT) [1.0000]
3.0010 (LOAD-ONE-BOX-ON-CARRIER B4 CAR R DEPOT PL2) [1.0000]
4.0012 (FILL-BOX-WITH-CONTENT B3 FOOD R DEPOT) [1.0000]
5.0015 (LOAD-ONE-BOX-ON-CARRIER B3 CAR R DEPOT PL4) [1.0000]
6.0017 (FILL-BOX-WITH-CONTENT B2 DRUGS R DEPOT) [1.0000]
7.0020 (LOAD-ONE-BOX-ON-CARRIER B2 CAR R DEPOT PL1) [1.0000]
8.0022 (MOVE-ROBOT-CARRIER R CAR DEPOT LOC1) [10.0000]
18.0025 (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR R LOC1 PL4) [1.0000]
19.0028 (UNLOAD-ONE-BOX-FROM-CARRIER B3 CAR R LOC1 PL2) [1.0000]
20.0030 (EMPTY-BOX-LEAVING-CONTENT B3 FOOD R P3 LOC1) [1.0000]
21.0033 (EMPTY-BOX-LEAVING-CONTENT B1 DRUGS R P1 LOC1) [1.0000]
22.0035 (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR R LOC1 PL3) [1.0000]
23.0037 (EMPTY-BOX-LEAVING-CONTENT B2 DRUGS R P2 LOC1) [1.0000]
24.0040 (UNLOAD-ONE-BOX-FROM-CARRIER B4 CAR R LOC1 PL1) [1.0000]
25.0042 (EMPTY-BOX-LEAVING-CONTENT B4 FOOD R P1 LOC1) [1.0000]

```

Lpg-td è risultato essere l'unico solver della libreria tra quelli testati a non terminare la memoria sull'istanza 2, sebbene sia stato necessario eseguirla in modalità speed. Il plan restituito è tuttavia lungi dall'essere ottimo, in maniera analoga a quanto accaduto con il planner custom.

Listing 19: **Lpg-td -speed** - Plan problem 2

```

; Version LPG-td-1.4

; Time 2158.07
; Search time 2158.07
; Parsing time 0.00
; Mutex time 0.00
; MetricValue 416.00

0.0002: (MOVE-ROBOT-CARRIER R1 CAR2 DEPOT LOC5) [10.0000]
0.0002: (LOAD-ONE-BOX-ON-CARRIER B3 CAR1 R2 DEPOT PL1CAR1)
[1.0000]
1.0005: (LOAD-ONE-BOX-ON-CARRIER B2 CAR1 R2 DEPOT PL2CAR1)
[1.0000]
2.0007: (MOVE-ROBOT-CARRIER R2 CAR1 DEPOT LOC2) [10.0000]
12.0010: (MOVE-ROBOT R2 LOC2 LOC5) [8.0000]
20.0012: (MOVE-ROBOT-CARRIER R2 CAR2 LOC5 DEPOT) [10.0000]

```

30.0015: (FILL-BOX-WITH-CONTENT B1 DRUGS R2 DEPOT) [1.0000]  
 31.0018: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
           [1.0000]  
 32.0020: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC5) [10.0000]  
 42.0023: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 LOC5 PL1CAR2)  
           [1.0000]  
 43.0025: (EMPTY-BOX-LEAVING-CONTENT B1 DRUGS R2 P6 LOC5)  
           [1.0000]  
 44.0028: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 LOC5 PL2CAR2)  
           [1.0000]  
 45.0030: (MOVE-ROBOT-CARRIER R2 CAR2 LOC5 DEPOT) [10.0000]  
 55.0033: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 DEPOT PL2CAR2)  
           [1.0000]  
 56.0035: (FILL-BOX-WITH-CONTENT B1 DRUGS R2 DEPOT) [1.0000]  
 57.0038: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
           [1.0000]  
 58.0040: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC4) [10.0000]  
 68.0043: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 LOC4 PL1CAR2)  
           [1.0000]  
 69.0045: (EMPTY-BOX-LEAVING-CONTENT B1 DRUGS R2 P5 LOC4)  
           [1.0000]  
 70.0048: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 LOC4 PL2CAR2)  
           [1.0000]  
 71.0050: (MOVE-ROBOT-CARRIER R2 CAR2 LOC4 DEPOT) [10.0000]  
 81.0053: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 DEPOT PL2CAR2)  
           [1.0000]  
 82.0055: (FILL-BOX-WITH-CONTENT B1 DRUGS R2 DEPOT) [1.0000]  
 83.0058: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
           [1.0000]  
 84.0060: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC3) [10.0000]  
 94.0063: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 LOC3 PL1CAR2)  
           [1.0000]  
 95.0065: (EMPTY-BOX-LEAVING-CONTENT B1 DRUGS R2 P4 LOC3)  
           [1.0000]  
 96.0068: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 LOC3 PL2CAR2)  
           [1.0000]  
 97.0070: (MOVE-ROBOT-CARRIER R2 CAR2 LOC3 DEPOT) [10.0000]  
 107.0073: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 DEPOT PL2CAR2)  
           [1.0000]  
 108.0075: (FILL-BOX-WITH-CONTENT B1 DRUGS R2 DEPOT) [1.0000]  
 109.0078: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
           [1.0000]  
 110.0080: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC1) [10.0000]  
 120.0083: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 LOC1 PL1CAR2)  
           [1.0000]  
 121.0086: (EMPTY-BOX-LEAVING-CONTENT B1 DRUGS R2 P2 LOC1)  
           [1.0000]

122.0088: (MOVE-ROBOT R2 LOC1 LOC2) [8.0000]  
 130.0090: (MOVE-ROBOT-CARRIER R2 CAR1 LOC2 DEPOT) [10.0000]  
 140.0093: (UNLOAD-ONE-BOX-FROM-CARRIER B3 CAR1 R2 DEPOT PL1CAR1)  
           [1.0000]  
 141.0095: (FILL-BOX-WITH-CONTENT B3 DRUGS R2 DEPOT) [1.0000]  
 142.0098: (LOAD-ONE-BOX-ON-CARRIER B3 CAR1 R2 DEPOT PL1CAR1)  
           [1.0000]  
 143.0100: (MOVE-ROBOT-CARRIER R2 CAR1 DEPOT LOC2) [10.0000]  
 153.0103: (UNLOAD-ONE-BOX-FROM-CARRIER B3 CAR1 R2 LOC2 PL1CAR1)  
           [1.0000]  
 154.0105: (MOVE-ROBOT-CARRIER R2 CAR1 LOC2 LOC5) [10.0000]  
 10.0005: (MOVE-ROBOT R1 LOC5 LOC2) [8.0000]  
 154.0105: (EMPTY-BOX-LEAVING-CONTENT B3 DRUGS R1 P3 LOC2)  
           [1.0000]  
 164.0107: (MOVE-ROBOT R2 LOC5 LOC1) [8.0000]  
 172.0110: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 LOC1 PL1CAR2)  
           [1.0000]  
 173.0112: (MOVE-ROBOT R2 LOC1 LOC5) [8.0000]  
 181.0115: (MOVE-ROBOT-CARRIER R2 CAR1 LOC5 DEPOT) [10.0000]  
 191.0117: (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR1 R2 DEPOT PL2CAR1)  
           [1.0000]  
 192.0120: (FILL-BOX-WITH-CONTENT B2 FOOD R2 DEPOT) [1.0000]  
 193.0122: (LOAD-ONE-BOX-ON-CARRIER B2 CAR1 R2 DEPOT PL1CAR1)  
           [1.0000]  
 194.0125: (MOVE-ROBOT-CARRIER R2 CAR1 DEPOT LOC3) [10.0000]  
 204.0127: (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR1 R2 LOC3 PL1CAR1)  
           [1.0000]  
 205.0129: (EMPTY-BOX-LEAVING-CONTENT B2 FOOD R2 P4 LOC3)  
           [1.0000]  
 206.0132: (MOVE-ROBOT R2 LOC3 LOC1) [8.0000]  
 214.0134: (MOVE-ROBOT-CARRIER R2 CAR2 LOC1 DEPOT) [10.0000]  
 224.0137: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
           [1.0000]  
 225.0139: (FILL-BOX-WITH-CONTENT B1 TOOLS R2 DEPOT) [1.0000]  
 226.0142: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL2CAR2)  
           [1.0000]  
 227.0144: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC5) [10.0000]  
 237.0146: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 LOC5 PL2CAR2)  
           [1.0000]  
 238.0149: (EMPTY-BOX-LEAVING-CONTENT B1 TOOLS R2 P6 LOC5)  
           [1.0000]  
 239.0151: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 LOC5 PL1CAR2)  
           [1.0000]  
 240.0154: (MOVE-ROBOT-CARRIER R2 CAR2 LOC5 DEPOT) [10.0000]  
 250.0156: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
           [1.0000]  
 251.0159: (FILL-BOX-WITH-CONTENT B1 FOOD R2 DEPOT) [1.0000]

252.0161: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL2CAR2)  
 [1.0000]  
 253.0164: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC5) [10.0000]  
 263.0166: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 LOC5 PL2CAR2)  
 [1.0000]  
 264.0168: (EMPTY-BOX-LEAVING-CONTENT B1 FOOD R2 P6 LOC5)  
 [1.0000]  
 265.0171: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 LOC5 PL1CAR2)  
 [1.0000]  
 266.0173: (MOVE-ROBOT-CARRIER R2 CAR2 LOC5 DEPOT) [10.0000]  
 276.0176: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
 [1.0000]  
 277.0178: (FILL-BOX-WITH-CONTENT B1 TOOLS R2 DEPOT) [1.0000]  
 278.0181: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL2CAR2)  
 [1.0000]  
 279.0183: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC4) [10.0000]  
 289.0186: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 LOC4 PL2CAR2)  
 [1.0000]  
 290.0188: (EMPTY-BOX-LEAVING-CONTENT B1 TOOLS R2 P5 LOC4)  
 [1.0000]  
 291.0190: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 LOC4 PL1CAR2)  
 [1.0000]  
 292.0193: (MOVE-ROBOT-CARRIER R2 CAR2 LOC4 DEPOT) [10.0000]  
 302.0195: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
 [1.0000]  
 303.0198: (FILL-BOX-WITH-CONTENT B1 TOOLS R2 DEPOT) [1.0000]  
 304.0200: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL2CAR2)  
 [1.0000]  
 305.0203: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC1) [10.0000]  
 315.0205: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 LOC1 PL2CAR2)  
 [1.0000]  
 316.0208: (EMPTY-BOX-LEAVING-CONTENT B1 TOOLS R2 P1 LOC1)  
 [1.0000]  
 317.0210: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 LOC1 PL1CAR2)  
 [1.0000]  
 318.0212: (MOVE-ROBOT-CARRIER R2 CAR2 LOC1 DEPOT) [10.0000]  
 328.0215: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R2 DEPOT PL1CAR2)  
 [1.0000]  
 329.0217: (FILL-BOX-WITH-CONTENT B1 FOOD R2 DEPOT) [1.0000]  
 330.0220: (LOAD-ONE-BOX-ON-CARRIER B1 CAR2 R2 DEPOT PL2CAR2)  
 [1.0000]  
 331.0222: (MOVE-ROBOT-CARRIER R2 CAR2 DEPOT LOC4) [10.0000]  
 155.0107: (MOVE-ROBOT R1 LOC2 LOC3) [8.0000]  
 206.0134: (LOAD-ONE-BOX-ON-CARRIER B2 CAR1 R1 LOC3 PL1CAR1)  
 [1.0000]  
 207.0137: (MOVE-ROBOT-CARRIER R1 CAR1 LOC3 LOC4) [10.0000]  
 341.0225: (MOVE-ROBOT-CARRIER R1 CAR2 LOC4 LOC1) [10.0000]

```

351.0227: (UNLOAD-ONE-BOX-FROM-CARRIER B1 CAR2 R1 LOC1 PL2CAR2)
          [1.0000]
352.0229: (EMPTY-BOX-LEAVING-CONTENT B1 FOOD R1 P1 LOC1)
          [1.0000]
341.0225: (MOVE-ROBOT-CARRIER R2 CAR1 LOC4 DEPOT) [10.0000]
351.0227: (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR1 R2 DEPOT PL1CAR1)
          [1.0000]
352.0229: (FILL-BOX-WITH-CONTENT B2 FOOD R2 DEPOT) [1.0000]
353.0232: (LOAD-ONE-BOX-ON-CARRIER B2 CAR1 R2 DEPOT PL2CAR1)
          [1.0000]
354.0234: (MOVE-ROBOT-CARRIER R2 CAR1 DEPOT LOC4) [10.0000]
364.0237: (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR1 R2 LOC4 PL2CAR1)
          [1.0000]
365.0239: (EMPTY-BOX-LEAVING-CONTENT B2 FOOD R2 P5 LOC4)
          [1.0000]

```

Infine, l'esecuzione dell'istanza 2 in modalità quality non è andata a buon fine in quanto il planner, seppur riuscendo a terminare, non è stato in grado di attuare la fase di miglioramento, restituendo, di fatto, un plan analogo al precedente.

Alla luce dei risultati ottenuti si è deciso di utilizzare il planner **Lpg-td** anche per lo svolgimento dell'esercizio di **temporal planning e robotics**.

### 4.3 Istanza 2 rilassata

Data l'impossibilità di ottenere un plan ottimale per l'istanza 2, si è deciso di creare una nuova istanza "rilassata" rimuovendo p4, p5 e p6 dall'istanza 2. Il planner custom riesce a risolvere questa istanza con peso dell'euristica pari a 1, restituendo un **plan ottimale**.

Listing 20: **Solver custom con peso dell'euristica unitario** - Plan problem 2 rilassato

```

problem instantiation done successfully (240 actions, 61 fluents)

```

```

* Leaving the Shire
* Found the Arkenstone :)
* Created nodes: 16120513
* Explored nodes: 1128270

```

found plan as follows:

```

00: (          fill-box-with-content b3 food r1 depot) [0]
01: (    load-one-box-on-carrier b3 car2 r1 depot pl21) [0]
02: (          fill-box-with-content b1 drugs r1 depot) [0]
03: (          fill-box-with-content b2 drugs r1 depot) [0]
04: (    load-one-box-on-carrier b1 car2 r1 depot pl22) [0]
05: (          move-robot-carrier r1 car2 depot loc1) [0]
06: (    load-one-box-on-carrier b2 car1 r2 depot pl12) [0]
07: (          move-robot-carrier r2 car1 depot loc2) [0]
08: ( unload-one-box-from-carrier b2 car1 r2 loc2 pl12) [0]

```

```

09: (      empty-box-leaving-content b2 drugs r2 p3 loc2) [0]
10: (      load-one-box-on-carrier b2 car1 r2 loc2 pl12) [0]
11: (              move-robot-carrier r2 car1 loc2 depot) [0]
12: ( unload-one-box-from-carrier b3 car2 r1 loc1 pl21) [0]
13: (      empty-box-leaving-content b3 food r1 p1 loc1) [0]
14: (unload-one-box-from-carrier b2 car1 r2 depot pl12) [0]
15: (              fill-box-with-content b2 tools r2 depot) [0]
16: ( unload-one-box-from-carrier b1 car2 r1 loc1 pl22) [0]
17: (      load-one-box-on-carrier b2 car1 r2 depot pl11) [0]
18: (      empty-box-leaving-content b1 drugs r1 p2 loc1) [0]
19: (              move-robot-carrier r2 car1 depot loc1) [0]
20: ( unload-one-box-from-carrier b2 car1 r1 loc1 pl11) [0]
21: (      empty-box-leaving-content b2 tools r1 p1 loc1) [0]

```

```

time spent:      0,04 seconds parsing
                  0,09 seconds encoding
                  65,59 seconds searching
                  65,72 seconds total time

```

```

memory used:      0,89 MBytes for problem representation
                  349,95 MBytes for searching
                  350,83 MBytes total

```

Si osservi come i due agenti robotici riescano a lavorare correttamente in parallelo, suddividendosi il lavoro.

## 5 TEMPORAL PLANNING AND ROBOTICS

Il terzo punto del progetto è suddiviso in una parte di temporal planning ed una di robotics planning.

### 5.1 Temporal planning

Viene richiesto di effettuare la conversione del dominio e della prima istanza descritta al punto precedente in PDDL 2.1, trasformando le azioni classiche in durative-actions. Viene inoltre richiesto di modellare il peso di ciascuno degli oggetti trasportati e di rendere il tempo di spostamento del robot con il carrier dipendente dal peso totale del carico.

#### 5.1.1 Conversione del file di dominio

È possibile usare come base il file prodotto dalla prima fase di conversione, effettuata per testare i diversi planner e descritta approfonditamente al punto 4.1. A tale versione è sufficiente aggiungere i fluents per modellare i pesi e modificare le azioni di conseguenza. In particolare sono state effettuate le seguenti modifiche:

- Sono stati aggiunti 3 fluents, **content-weight**, **box-weight** e **carrier-weight**, che modellano rispettivamente il peso di un content (definito nel problema), il peso

di un box (dipendente dal content con cui è riempito) e il peso totale trasportato dal carrier (dipendente da quello dei box caricati).

```
(:functions
  (path-cost ?r - robot)

  (content-weight ?c - content)
  (box-weight ?b - box)
  (carrier-weight ?car - carrier)
)
```

- Alla fine dell'azione **fill-box-with-content** viene aggiornato il peso del box, assegnandogli il peso corrispondente al content.

```
(:durative-action fill-box-with-content
:parameters
  (?b - box
   ?c - content
   ?r - robot
   ?loc -location
  )

:duration(= ?duration 1)

:condition
  (and
    (at start (box-empty ?b))
    (over all (at ?c ?loc))
    (over all (at ?b ?loc))
    (over all (at ?r ?loc))
    (at start (robot-free ?r))
  )
:effect
  (and
    (at start (not (robot-free ?r)))
    (at end (robot-free ?r))
    (at end (box-contains ?b ?c))
    (at start (not (box-empty ?b)))

    (at end (assign (box-weight ?b) (content-weight ?c)))
    (at end (increase (path-cost ?r) 1))
  )
)
```

- In maniera analoga, alla fine dell'operazione di **empty-box-leaving-content** viene azzerato il peso del box.

```
(:durative-action empty-box-leaving-content
:parameters
```

```

    (?b - box
     ?c - content
     ?r - robot
     ?p - person
     ?loc -location
    )

    :duration(= ?duration 1)

    :condition
      (and
        (at start (box-contains ?b ?c))
        (over all (at ?b ?loc))
        (over all (at ?r ?loc))
        (over all (at ?p ?loc))
        (over all (person-needs-content ?p ?c))
        (at start (robot-free ?r))
      )
    :effect
      (and
        (at start (not (robot-free ?r)))
        (at end (robot-free ?r))
        (at start (not (box-contains ?b ?c)))
        (at end (box-empty ?b))
        (at end (person-has-content ?p ?c))

        (at end (assign (box-weight ?b) 0))
        (at end (increase (path-cost ?r) 1))
      )
  )

```

- Al termine dell'azione **load-one-box-on-carrier** viene incrementato il peso trasportato dal carrier in base al peso del box caricato.

```

(:durative-action load-one-box-on-carrier
  :parameters
    (?b - box
     ?car - carrier
     ?r - robot
     ?loc -location
     ?pl - place
    )

  :duration(= ?duration 1)

  :condition
    (and
      (at start (at ?b ?loc))

```



```

        (over all (at ?car ?loc))
        (over all (at ?r ?loc))
        (over all (place-of-carrier ?pl ?car))
        (at start (place-available ?pl))
        (at start (robot-free ?r))
    )

:effect
    (and
        (at start (not (robot-free ?r)))
        (at end (robot-free ?r))
        (at start (not (at ?b ?loc)))
        (at end (box-loaded-on-carrier ?b ?car))
        (at start (not (place-available ?pl)))
        (at start (place-occupied ?pl))

        (at end (increase (carrier-weight ?car) (
            box-weight ?b)))
        (at end (increase (path-cost ?r) 1))
    )
)

```

- Similmente, al termine dell'azione **unload-one-box-from-carrier** il peso trasportato dal carrier viene decrementato.

```

(:durative-action unload-one-box-from-carrier
:parameters
    (?b - box
     ?car - carrier
     ?r - robot
     ?loc -location
     ?pl - place
    )

:duration(= ?duration 1)

:condition
    (and
        (over all (at ?car ?loc))
        (over all (at ?r ?loc))
        (at start (box-loaded-on-carrier ?b ?car))
        (over all (place-of-carrier ?pl ?car))
        (over all (place-occupied ?pl))
        (at start (robot-free ?r))
    )

:effect
    (and

```

```

(at start (not (robot-free ?r)))
(at end (robot-free ?r))
(at end (at ?b ?loc))
(at start (not (box-loaded-on-carrier ?b ?car)))
(at end (not (place-occupied ?pl)))
(at end (place-available ?pl))

(at end (decrease (carrier-weight ?car) (
  box-weight ?b)))
(at end (increase (path-cost ?r) 1))
)
)

```

- La durata dello spostamento del solo robot è stata ridotta a 4, mentre la durata dello spostamento del robot con il carrier è adesso ottenuta come  $4 * (\text{peso trasportato})$ .

Le modifiche alla durata delle operazioni sono state propagate anche al fluent **path-cost**, che sarà usato successivamente come metrica al fine di minimizzare il costo del plan restituito.

```

(:durative-action move-robot-carrier
  :parameters
    (?r - robot
     ?car - carrier
     ?from - location
     ?to - location
    )

  :duration(= ?duration (* (carrier-weight ?car) 4))

  :condition
    (and
      (at start (at ?r ?from))
      (at start (at ?car ?from))
      (at start (robot-free ?r))
    )

  :effect
    (and
      (at start (not (robot-free ?r)))
      (at end (robot-free ?r))
      (at start (not (at ?r ?from)))
      (at end (at ?r ?to))
      (at start (not (at ?car ?from)))
      (at end (at ?car ?to))

      (at end (increase (path-cost ?r) (* (
        carrier-weight ?car) 4)))
    )
)

```

```
)
)
```

### 5.1.2 Conversione del file di problema

Si è reso necessario convertire anche il file di problema per adattarlo alle modifiche effettuate, definendo i valori iniziali dei fluents utilizzati:

- Il peso di drugs, food e tools è stato fissato rispettivamente a 1, 2 e 3, come indicato dalla traccia.
- Il peso dei box è stato inizialmente fissato a 0, coerentemente con lo stato di “box-empty”.
- Il peso iniziale del carrier è stato fissato a 1. Tale vincolo è necessario al fine di evitare che spostamenti con carrier vuoto o di soli box vuoti abbiano costo nullo.

Segue il file di problema così modificato.

Listing 21: **weighted-problem1.pddl**

```
(define
  (problem deliveringcontents)
  (:domain emergencyServicesLogistics)

  (:objects
    depot loc1 - location
    b1 b2 b3 b4 b5 - box
    p1 p2 p3 - person
    r - robot
    car - carrier
    pl1 pl2 pl3 pl4 - place

    drugs food tools - content
  )

  (:init
    (= (content-weight drugs) 1)
    (= (content-weight food) 2)
    (= (content-weight tools) 3)

    (= (box-weight b1) 0)
    (= (box-weight b2) 0)
    (= (box-weight b3) 0)
    (= (box-weight b4) 0)
    (= (box-weight b5) 0)

    (= (carrier-weight car) 1)

    (= (path-cost r) 0)
```

```

(at b1 depot)
(at b2 depot)
(at b3 depot)
(at b4 depot)
(at b5 depot)

(at r depot)
(at car depot)

(robot-free r)

(at food depot)
(at drugs depot)
(at tools depot)

(at p1 loc1)
(at p2 loc1)
(at p3 loc1)

(person-needs-content p1 food)
(person-needs-content p1 drugs)

(person-needs-content p2 drugs)

(person-needs-content p3 food)

(box-empty b1)
(box-empty b2)
(box-empty b3)
(box-empty b4)
(box-empty b5)

(place-available pl1)
(place-available pl2)
(place-available pl3)
(place-available pl4)

(place-of-carrier pl1 car)
(place-of-carrier pl2 car)
(place-of-carrier pl3 car)
(place-of-carrier pl4 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))
)

```

### 5.1.3 Risoluzione e ottenimento del plan

Utilizzando il solver **Lpg-td** in modalità **-quality** si è ottenuto il seguente plan ottimale.

Listing 22: Plan problem 1 con pesi

```

; Version LPG-td-1.4

; Time 13.01
; Plan generation time 2.81
; Search time 2.80
; Parsing time 0.00
; Mutex time 0.00
; MetricValue 44.00

0.0003: (FILL-BOX-WITH-CONTENT B3 DRUGS R DEPOT) [1.0000]
1.0005: (FILL-BOX-WITH-CONTENT B2 FOOD R DEPOT) [1.0000]
2.0008: (FILL-BOX-WITH-CONTENT B5 FOOD R DEPOT) [1.0000]
3.0010: (LOAD-ONE-BOX-ON-CARRIER B2 CAR R DEPOT PL4) [1.0000]
4.0012: (FILL-BOX-WITH-CONTENT B4 DRUGS R DEPOT) [1.0000]
5.0015: (LOAD-ONE-BOX-ON-CARRIER B4 CAR R DEPOT PL3) [1.0000]
6.0017: (LOAD-ONE-BOX-ON-CARRIER B5 CAR R DEPOT PL2) [1.0000]
7.0020: (LOAD-ONE-BOX-ON-CARRIER B3 CAR R DEPOT PL1) [1.0000]
8.0022: (MOVE-ROBOT-CARRIER R CAR DEPOT LOC1) [28.0000]
36.0025: (UNLOAD-ONE-BOX-FROM-CARRIER B3 CAR R LOC1 PL4)
[1.0000]
37.0028: (UNLOAD-ONE-BOX-FROM-CARRIER B5 CAR R LOC1 PL3)
[1.0000]
38.0030: (EMPTY-BOX-LEAVING-CONTENT B5 FOOD R P3 LOC1) [1.0000]
39.0033: (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR R LOC1 PL1)
[1.0000]
40.0035: (UNLOAD-ONE-BOX-FROM-CARRIER B4 CAR R LOC1 PL2)
[1.0000]
41.0037: (EMPTY-BOX-LEAVING-CONTENT B4 DRUGS R P1 LOC1) [1.0000]
42.0040: (EMPTY-BOX-LEAVING-CONTENT B3 DRUGS R P2 LOC1) [1.0000]
43.0042: (EMPTY-BOX-LEAVING-CONTENT B2 FOOD R P1 LOC1) [1.0000]

```

## 5.2 Robotics planning

Il task di robotics planning richiede di implementare il problema del punto precedente in **PlanSys2**, utilizzando delle **fake action** e sfruttando il plan appena generato.

### 5.2.1 Definizione delle fake action

La definizione delle fake action avviene mediante file C++, in particolare uno per azione. Tali file devono essere inseriti all'interno della sottocartella `src` dentro la cartella del progetto. Tutte le azioni seguono il template di fake action descritto nel tutorial di PlanSys2, pertanto ne sarà descritta dettagliatamente solo una, a titolo esemplificativo.

```
1 #include <vector>
2 #include <string>
3 #include <memory>
4 #include <algorithm>
5 #include "plansys2_executor/ActionExecutorClient.hpp"
6 #include "rclcpp/rclcpp.hpp"
7 #include "rclcpp_action/rclcpp_action.hpp"
8
9 using namespace std::chrono_literals;
10
11 class LoadOneBoxOnCarrierAction : public plansys2::
    ActionExecutorClient
12 {
13 public:
14     LoadOneBoxOnCarrierAction()
15     : plansys2::ActionExecutorClient("load_one_box_on_carrier", 200
        ms)
16     {
17         progress_ = 0.0;
18     }
19
20 private:
21     void do_work()
22     {
23         std::vector<std::string> arguments = get_arguments();
24         // box carrier robot location place
25
26         if (progress_ < 1.0) {
27             progress_ += 0.2;
28             send_feedback(progress_, "Robot_" + arguments[2] + "_loading_box
                _" + arguments[0] + "_on_carrier_" + arguments[1] + "_at_" +
                arguments[3] + "_using_place_" + arguments[4]);
29         } else {
30             finish(true, 1.0, "Robot_" + arguments[2] + "_loaded_box_" +
                arguments[0] + "_on_carrier_" + arguments[1] + "_at_" +
                arguments[3] + "_using_place_" + arguments[4]);
31             progress_ = 0.0;
```

```

32     std::cout << std::endl;
33 }
34
35 std::cout << "\r\e[K" << std::flush;
36 std::cout << "Robot_" + arguments[2] + "_loading_box_" + arguments
    [0] + "_on_carrier_" + arguments[1] + "_at_" + arguments[3] + "_
    using_place_" + arguments[4] + "..._" << std::min(100.0,
    progress_ * 100.0) << "%]_" <<
37 std::flush;
38 }
39
40 float progress_;
41 };
42
43 int main(int argc, char ** argv)
44 {
45     rclcpp::init(argc, argv);
46     auto node = std::make_shared<LoadOneBoxOnCarrierAction>();
47     node->set_parameter(rclcpp::Parameter("action_name", "LOAD-ONE-
        BOX-ON-CARRIER"));
48     node->trigger_transition(lifecycle_msgs::msg::Transition::
        TRANSITION_CONFIGURE);
49     rclcpp::spin(node->get_node_base_interface());
50     rclcpp::shutdown();
51
52     return 0;
53 }

```

Segue la descrizione dei punti salienti del codice:

- Mediante l'istruzione **ActionExecutorClient** di linea 15 vengono indicati il nome del nodo che eseguirà l'azione fake e ogni quanto la funzione **do\_work()** verrà invocata, in questo caso 200 millisecondi.
- La funzione **do\_work()** è quella che modella l'azione: al suo interno viene definito l'incremento dell'indicatore di progresso e si inviano i feedback sullo stato dell'esecuzione. In particolare, l'incremento è stato scelto in modo da far corrispondere i tempi delle azioni fake con i tempi indicati nel plan che sarà eseguito. Inoltre, le stampe ed i feedback presenti sono stati arricchiti con informazioni relative agli oggetti coinvolti.
- Il parametro **action-name** di linea 47 indica l'azione del plan a cui corrisponderà l'invocazione dell'esecutore.

### 5.2.2 Configurazione dei file CMakeLists e launch

Dopo aver definito le fake action è necessario configurare opportunamente il file **CMakeList** presente all'interno della directory di progetto e il file python di **launch** presente nell'omonima sottocartella, indicando le fake action da associare ad ogni nodo.

Listing 23: File **CMakeList**

```

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(move_robot_action_node src/move_robot_action_node.
  cpp)
ament_target_dependencies(move_robot_action_node ${dependencies})

add_executable(move_robot_carrier_action_node src/
  move_robot_carrier_action_node.cpp)
ament_target_dependencies(move_robot_carrier_action_node ${
  dependencies})

add_executable(fill_box_with_content_action_node src/
  fill_box_with_content_action_node.cpp)
ament_target_dependencies(fill_box_with_content_action_node ${
  dependencies})

add_executable(empty_box_leaving_content_action_node src/
  empty_box_leaving_content_action_node.cpp)
ament_target_dependencies(empty_box_leaving_content_action_node ${
  dependencies})

add_executable(load_one_box_on_carrier_action_node src/
  load_one_box_on_carrier_action_node.cpp)
ament_target_dependencies(load_one_box_on_carrier_action_node ${
  dependencies})

add_executable(unload_one_box_from_carrier_action_node src/

```



```

        unload_one_box_from_carrier_action_node.cpp)
ament_target_dependencies(unload_one_box_from_carrier_action_node
    ${dependencies})

install(DIRECTORY launch pddl DESTINATION share/${PROJECT_NAME})

install(TARGETS
    move_robot_action_node
    move_robot_carrier_action_node
    fill_box_with_content_action_node
    empty_box_leaving_content_action_node
    load_one_box_on_carrier_action_node
    unload_one_box_from_carrier_action_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()

```

Listing 24: File **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/weighted/
        weighted_domain.pddl',
        'namespace': namespace
    }.items())

# Specify the actions
move_robot_cmd = Node(
    package='plansys2_project',
    executable='move_robot_action_node',
    name='move_robot_action_node',
    namespace=namespace,
    output='screen',
    parameters=[])

move_robot_carrier_cmd = Node(
    package='plansys2_project',
    executable='move_robot_carrier_action_node',
    name='move_robot_carrier_action_node',
    namespace=namespace,
    output='screen',
    parameters=[])

fill_box_cmd = Node(
    package='plansys2_project',
    executable='fill_box_with_content_action_node',
    name='fill_box_with_content_action_node',
    namespace=namespace,
    output='screen',
    parameters=[])

empty_box_cmd = Node(
    package='plansys2_project',

```

```

        executable='empty_box_leaving_content_action_node',
        name='empty_box_leaving_content_action_node',
        namespace=namespace,
        output='screen',
        parameters=[])

load_cmd = Node(
    package='plansys2_project',
    executable='load_one_box_on_carrier_action_node',
    name='load_one_box_on_carrier_action_node',
    namespace=namespace,
    output='screen',
    parameters=[])

unload_cmd = Node(
    package='plansys2_project',
    executable='unload_one_box_from_carrier_action_node',
    name='unload_one_box_from_carrier_action_node',
    namespace=namespace,
    output='screen',
    parameters=[])

# Create the launch description and populate
ld = LaunchDescription()

ld.add_action(declare_namespace_cmd)

# Declare the launch options
ld.add_action(plansys2_cmd)

ld.add_action(move_robot_cmd)
ld.add_action(move_robot_carrier_cmd)
ld.add_action(fill_box_cmd)
ld.add_action(empty_box_cmd)
ld.add_action(load_cmd)
ld.add_action(unload_cmd)

return ld

```

### 5.2.3 Inizializzazione dell'ambiente di esecuzione e simulazione

Una volta definiti tutti i file necessari all'esecuzione è necessario effettuare la build e l'inizializzazione dell'ambiente di esecuzione eseguendo le istruzioni:

```

colcon build --symlink-install
source install/setup.bash

```

Successivamente si lancia PlanSys2 da un terminale mediante l'istruzione:

```
ros2 launch plansys2_project plansys2_project_launch.py
```

Da un altro terminale si avvia il **terminale PlanSys2** con l'istruzione:

```
ros2 run plansys2_terminal plansys2_terminal
```

PlanSys2 non supporta direttamente la definizione delle istanze di problema mediante i file PDDL, pertanto è necessario usare i comandi specifici della console PlanSys2. Tali comandi vanno a definire un'istanza di problema del tutto equivalente a quella indicata mediante il file **weighted-problem1.pddl** discusso precedentemente. Per comodità di esecuzione tutti i comandi sono stati inseriti in un file **commands**, eseguibile direttamente con l'istruzione:

```
source /home/aiguy/plansys2_ws/src /  
ros2_planning_system_examples/plansys2_project/launch/  
commands
```

Listing 25: File **commands**

```
set instance DEPOT location  
set instance LOC1 location  
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 PL1 place  
set instance PL2 place  
set instance PL3 place  
set instance PL4 place  
set instance DRUGS content  
set instance FOOD content  
set instance TOOLS content  
  
set function (= (content-weight DRUGS) 1)  
set function (= (content-weight FOOD) 2)  
set function (= (content-weight TOOLS) 3)  
  
set function (= (box-weight B1) 0)  
set function (= (box-weight B2) 0)  
set function (= (box-weight B3) 0)
```

```

set function (= (box-weight B4) 0)
set function (= (box-weight B5) 0)

set function (= (carrier-weight CAR) 1)

set function (= (path-cost R) 0)


set predicate (at B1 DEPOT)
set predicate (at B2 DEPOT)
set predicate (at B3 DEPOT)
set predicate (at B4 DEPOT)
set predicate (at B5 DEPOT)

set predicate (at R DEPOT)
set predicate (at CAR DEPOT)

set predicate (robot-free R)

set predicate (at FOOD DEPOT)
set predicate (at DRUGS DEPOT)
set predicate (at TOOLS DEPOT)

set predicate (at P1 LOC1)
set predicate (at P2 LOC1)
set predicate (at P3 LOC1)

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 (box-empty B1)
set predicate (box-empty B2)
set predicate (box-empty B3)
set predicate (box-empty B4)
set predicate (box-empty B5)

set predicate (place-available PL1)
set predicate (place-available PL2)
set predicate (place-available PL3)
set predicate (place-available PL4)

set predicate (place-of-carrier PL1 CAR)
set predicate (place-of-carrier PL2 CAR)

```

set predicate	(place-of-carrier PL3 CAR)
set predicate	(place-of-carrier PL4 CAR)

A questo punto arrivati è sufficiente avviare la simulazione sul piano precedentemente generato con Lpg-td, mediante l'istruzione

```
run plan-file /home/aiguy/plansys2_ws/src/  
ros2_planning_system_examples/plansys2_project/pddl/  
weighted/plansys2_plan_1.plan
```

La simulazione viene completata con successo e gli indicatori di progresso delle azioni avanzano come previsto, emulando i tempi indicati nel plan.

Si riportano gli screenshot degli output dell'esecuzione e, per comodità di consultazione, gli stessi output in formato testuale.

```

aiguy@ubu22:~/plansys2_ws$ ros2 run plansys2_terminal plansys2_terminal
[INFO] [1689351318.756659046] [terminal]: No problem file specified.
ROS2 Planning System console. Type "quit" to finish
> source /home/aiguy/plansys2_ws/src/ros2_planning_system_examples/plansys2_project/launch/commands
done
done
done
done
done
done
done
done
done
done
> run plan-file /home/aiguy/plansys2_ws/src/ros2_planning_system_examples/plansys2_project/pddl/weighted/plansys2_plan_1.plan
The plan read from "/home/aiguy/plansys2_ws/src/ros2_planning_system_examples/plansys2_project/pddl/weighted/plansys2_plan_1.plan" is
0.0003: (FILL-BOX-WITH-CONTENT B3 DRUGS R DEPOT) [1]
1.0005: (FILL-BOX-WITH-CONTENT B2 FOOD R DEPOT) [1]
2.0008: (FILL-BOX-WITH-CONTENT B5 FOOD R DEPOT) [1]
3.001: (LOAD-ONE-BOX-ON-CARRIER B2 CAR R DEPOT PL4) [1]
4.0012: (FILL-BOX-WITH-CONTENT B4 DRUGS R DEPOT) [1]
5.0015: (LOAD-ONE-BOX-ON-CARRIER B4 CAR R DEPOT PL3) [1]
6.0017: (LOAD-ONE-BOX-ON-CARRIER B5 CAR R DEPOT PL2) [1]
7.002: (LOAD-ONE-BOX-ON-CARRIER B3 CAR R DEPOT PL1) [1]
8.0022: (MOVE-ROBOT-CARRIER R CAR DEPOT LOC1) [28]
36.0025: (UNLOAD-ONE-BOX-FROM-CARRIER B3 CAR R LOC1 PL4) [1]
37.0028: (UNLOAD-ONE-BOX-FROM-CARRIER B5 CAR R LOC1 PL3) [1]
38.003: (EMPTY-BOX-LEAVING-CONTENT B5 FOOD R P3 LOC1) [1]
39.0033: (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR R LOC1 PL1) [1]
40.0035: (UNLOAD-ONE-BOX-FROM-CARRIER B4 CAR R LOC1 PL2) [1]
41.0037: (EMPTY-BOX-LEAVING-CONTENT B4 DRUGS R P1 LOC1) [1]
42.004: (EMPTY-BOX-LEAVING-CONTENT B3 DRUGS R P2 LOC1) [1]
43.0042: (EMPTY-BOX-LEAVING-CONTENT B2 FOOD R P1 LOC1) [1]
[INFO] [1689351365.248053177] [executor_client]: Plan Succeeded

Successful finished
>

```

Figura 5.1: Terminale PlanSys2

```
> run plan-file /home/aiguy/plansys2_ws/src/  
  ros2_planning_system_examples/plansys2_project/pddl/weighted/  
  plansys2_plan_1.plan
```

The plan read from "/home/aiguy/plansys2\_ws/src/  
 ros2\_planning\_system\_examples/plansys2\_project/pddl/weighted/  
 plansys2\_plan\_1.plan" is

```
0.0003: (FILL-BOX-WITH-CONTENT B3 DRUGS R DEPOT)          [1]  
1.0005: (FILL-BOX-WITH-CONTENT B2 FOOD R DEPOT) [1]  
2.0008: (FILL-BOX-WITH-CONTENT B5 FOOD R DEPOT) [1]  
3.001:  (LOAD-ONE-BOX-ON-CARRIER B2 CAR R DEPOT PL4)      [1]  
4.0012: (FILL-BOX-WITH-CONTENT B4 DRUGS R DEPOT)          [1]  
5.0015: (LOAD-ONE-BOX-ON-CARRIER B4 CAR R DEPOT PL3)      [1]  
6.0017: (LOAD-ONE-BOX-ON-CARRIER B5 CAR R DEPOT PL2)      [1]  
7.002:  (LOAD-ONE-BOX-ON-CARRIER B3 CAR R DEPOT PL1)      [1]  
8.0022: (MOVE-ROBOT-CARRIER R CAR DEPOT LOC1) [28]  
36.0025: (UNLOAD-ONE-BOX-FROM-CARRIER B3 CAR R LOC1 PL4)  [1]  
37.0028: (UNLOAD-ONE-BOX-FROM-CARRIER B5 CAR R LOC1 PL3)  [1]  
38.003: (EMPTY-BOX-LEAVING-CONTENT B5 FOOD R P3 LOC1) [1]  
39.0033: (UNLOAD-ONE-BOX-FROM-CARRIER B2 CAR R LOC1 PL1)  [1]  
40.0035: (UNLOAD-ONE-BOX-FROM-CARRIER B4 CAR R LOC1 PL2)  [1]  
41.0037: (EMPTY-BOX-LEAVING-CONTENT B4 DRUGS R P1 LOC1) [1]  
42.004: (EMPTY-BOX-LEAVING-CONTENT B3 DRUGS R P2 LOC1) [1]  
43.0042: (EMPTY-BOX-LEAVING-CONTENT B2 FOOD R P1 LOC1) [1]
```

```
[INFO] [1689351365.248053177] [executor_client]: Plan Succeeded
```

Successful finished

```
>
```

```

Robot R filling box B3 with content DRUGS at DEPOT... [100%]
Robot R filling box B2 with content FOOD at DEPOT... [100%]
Robot R filling box B5 with content FOOD at DEPOT... [100%]
Robot R loading box B2 on carrier CAR at DEPOT using place PL4... [100%]
Robot R filling box B4 with content DRUGS at DEPOT... [100%]
Robot R loading box B4 on carrier CAR at DEPOT using place PL3... [100%]
Robot R loading box B5 on carrier CAR at DEPOT using place PL2... [100%]
Robot R loading box B3 on carrier CAR at DEPOT using place PL1... [100%]
Robot R moving carrier CAR from DEPOT to LOC1... [100%]
Robot R unloading box B3 from carrier CAR at LOC1 leaving place PL4... [100%]
Robot R unloading box B5 from carrier CAR at LOC1 leaving place PL3... [100%]
Robot R emptying box B5 delivering content FOOD to P3 at LOC1... [100%]
Robot R unloading box B2 from carrier CAR at LOC1 leaving place PL1... [100%]
Robot R unloading box B4 from carrier CAR at LOC1 leaving place PL2... [100%]
Robot R emptying box B4 delivering content DRUGS to P1 at LOC1... [100%]
Robot R emptying box B3 delivering content DRUGS to P2 at LOC1... [100%]
Robot R emptying box B2 delivering content FOOD to P1 at LOC1... [100%]
[plansys2_node-1] [INFO] [1689351362.981995556] [executor]: Plan Succeeded

```

Figura 5.2: Finestra di lancio

```

Robot R filling box B3 with content DRUGS at DEPOT... [100%]
Robot R filling box B2 with content FOOD at DEPOT... [100%]
Robot R filling box B5 with content FOOD at DEPOT... [100%]
Robot R loading box B2 on carrier CAR at DEPOT using place PL4...
[100%]
Robot R filling box B4 with content DRUGS at DEPOT... [100%]
Robot R loading box B4 on carrier CAR at DEPOT using place PL3...
[100%]
Robot R loading box B5 on carrier CAR at DEPOT using place PL2...
[100%]
Robot R loading box B3 on carrier CAR at DEPOT using place PL1...
[100%]
Robot R moving carrier CAR from DEPOT to LOC1... [100%]
Robot R unloading box B3 from carrier CAR at LOC1 leaving place
PL4... [100%]
Robot R unloading box B5 from carrier CAR at LOC1 leaving place
PL3... [100%]
Robot R emptying box B5 delivering content FOOD to P3 at LOC1...
[100%]
Robot R unloading box B2 from carrier CAR at LOC1 leaving place
PL1... [100%]
Robot R unloading box B4 from carrier CAR at LOC1 leaving place
PL2... [100%]
Robot R emptying box B4 delivering content DRUGS to P1 at LOC1...
[100%]
Robot R emptying box B3 delivering content DRUGS to P2 at LOC1...
[100%]
Robot R emptying box B2 delivering content FOOD to P1 at LOC1...
[100%]

```



```
[plansys2_node-1] [INFO] [1689351362.981995556] [executor]: Plan  
Succeeded
```