Healthcare Planning System

Assignment for the course Automated Planning Theory and Practice

Luca Cazzola
Artificial Intelligence Systems (a.y. 2024-2025)
University of Trento (Italy)
luca.cazzola-1@studenti.unitn.it - 248716

***Abstract*—This work explores planning techniques through a healthcare facility logistics scenario involving robotic agents for supply delivery and patient escort. The study progressively develops planning models using PDDL (Planning Domain Definition Language) and HDDL (Hierarchical Domain Definition Language), advancing from basic formulations to carrier-enhanced transportation, hierarchical decomposition, temporal planning, and ROS2 integration. Results demonstrate successful implementation across multiple planning paradigms, achieving optimal solutions for smaller instances and efficient sub-optimal solutions for complex scenarios. The work culminates in functional ROS2 integration, bridging theoretical planning research with practical robotic applications.**

***Index Terms*—PDDL, HDDL, planning, healthcare logistics, robotic coordination, temporal planning, hierarchical planning, ROS2, multi-agent systems.**

I. Introduction

Planning plays an important role in artificial intelligence applications where systems must coordinate complex sequences of actions to achieve desired outcomes. In healthcare environments, efficient coordination of resources and patient flow presents complex logistical challenges that can benefit from planning approaches. This assignment explores these challenges by developing comprehensive planning models for healthcare facility management, focusing on robotic agents responsible for medical supply distribution and patient escort services.

The healthcare planning domain provides an excellent case study for exploring various planning methodologies covered in the course. The environment features interconnected locations forming a facility roadmap, specialized robotic agents with distinct capabilities, medical units with specific requirements, and coordination constraints that must be satisfied while optimizing performance. These characteristics allow for the investigation of different planning paradigms and their effectiveness in real-world logistics scenarios.

This assignment is structured as follows: sec. II describes in more detail the domain, sec. III presents the problem formulation and modeling approaches, progressing from basic PDDL formulations through carrier-enhanced transportation, hierarchical task decomposition, temporal planning, and ROS2 integration. sec. IV provides comprehensive experimental results and performance analysis across all modeling approaches. sec. V concludes with key findings and observations. Each modeling approach builds upon previous foundations while addressing specific aspects of the healthcare logistics challenge, demonstrating the practical application of theoretical planning concepts studied throughout the course. All the code is available on GitHub.

II. Problem Understanding

This healthcare facility planning scenario focuses on developing strategies for robotic agents (bots) to efficiently deliver medical supplies and escort patients throughout a facility with multiple medical units at various locations. The following actors operate within this system along with their respective operational constraints:

- **Location**: Represents a geographical spot within the healthcare facility. All locations are connected through a predefined roadmap, which creates a graph-like topology that defines allowed movement paths throughout the facility.
- **Medical Unit**: Healthcare service providers that operate within the facility and require specific medical supplies, patients, or both. Each medical unit is positioned at a designated location, with multiple units potentially sharing the same location.
- **Supply**: Individual pieces of medical equipment such as `scalpel`, `aspirin`, or `tongue depressor` that medical units require for operations. Supplies cannot move independently between locations and must be transported within boxes, with each box accommodating exactly one supply item.
- **Box**: Storage containers designed for transporting medical supplies throughout the facility. Each box starts at a specific location and can hold one supply item. Boxes can be filled with supplies or emptied to make their contents available at the current location.
- **Patient**: Healthcare facility clients who require medical treatment and assistance in reaching appropriate medical units. Patients depend on helper bots for navigation and move alongside their assigned robotic agent during escort operations.
- **Worker Bot**: Specialized robotic agents responsible for supply management and logistics. These bots can fill and empty boxes, pick up boxes from their current location, navigate the facility roadmap while transporting boxes, and deliver supplies to satisfy medical unit requirements.

Their carrying capacity defaults to one box but can be enhanced through carrier ownership when available.

- **Helper Bot**: Specialized robotic agents dedicated to patient assistance and guidance. These bots escort patients to their designated medical units, navigate the facility roadmap while accompanying patients, and ensure successful patient delivery to appropriate healthcare services.
- **Carrier**: Transportation enhancement equipment (available from sec. III-B onwards) that increases a worker bot's carrying capacity. Each carrier has a predetermined capacity limit that sets how many boxes can be transported simultaneously, allowing worker bots to exceed the standard single-box limitation.

A. Initial State

For convenience, the healthcare facility topology remains fixed across all experiments, with medical units also remaining consistent and positioned at the same locations throughout different problems. This simplifies understanding solutions quality, while problem complexity can be increasing by modifying other, more relevant components, like the number of bots, boxes and supplies. The chosen structure is illustrated at fig. 1 and built to be sufficiently large to avoid over-simplicity.

When planning starts, the system is set as follows: All patients and helper bots begin at the `entrance` location, while all worker bots start at the `central_warehouse`, which serves as the primary logistic point in the healthcare facility. Both boxes and supplies are initially positioned at the `central_warehouse`, with all boxes beginning in an empty state and all robotic agents available for deployment, meaning they're not carrying any box or assisting patients.

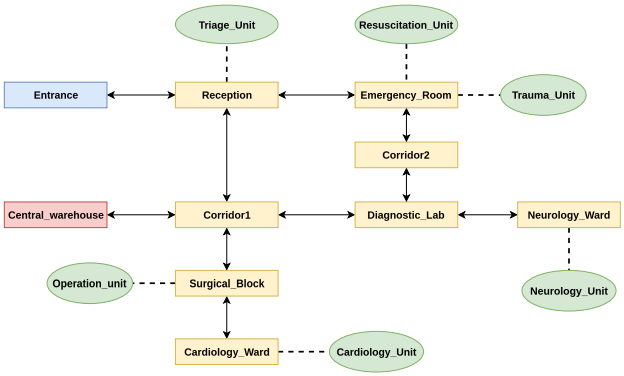

Fig. 1: The healthcare facility roadmap. Rectangles represent locations, while ellipses denote medical units. Solid arrow lines indicate connectivity between locations (all bidirectional), and dashed lines represent the relationships between locations and medical units. When a medical unit is connected to a location, it signifies that the medical unit is situated at that particular location.

B. Goal

The goal involves ensuring specific medical units receive required supplies and patients reach their designated units. In some problem evaluations, an additional goal is added to "restore initial state", meaning that after all supplies are delivered to medical units and patients are being taken care of at the correct units, all boxes should return empty to `central_warehouse` along with worker bots, while helper bots should all return to `entrance`, simulating that the system is ready for future tasks.

C. Optimal planning

Given the above problem formulation we can identify what's the expected optimal planning output from the system. Let's assume a very simple scenario with 1 worker bot, 1 helper bot, a healthcare facility structured as in fig. 1, 1 box available, 1 patient, `reanimation_unit` in need of supply `defibrillator` and the patient needing to be treated at `cardiology_unit`. Here follows the expected outcome:

1) **Supply delivery**: The worker bot is expected to: Fill the box with `defibrillator` → pick up the box → route towards `reanimation_unit` respecting the given roadmap and choosing the shortest path possible → drop the box → perform delivery. In case of carrier presence, the procedure should be the same, but with the possibility of loading multiple boxes if fewer action-steps are performed doing so. For instance, a robot might fill and load 2 boxes and deliver them before returning to the central warehouse if it's more efficient.
2) **Patient delivery**: The helper bot should follow a very similar schema to that of worker bots. An optimal helper bot is expected to: pick up the patient → route towards `cardiology_unit` using the shortest path available and respecting the roadmap → drop the patient → deliver them to the medical unit.

III. Modeling

Given the progressive structure of the assignment, early design decisions significantly impact system quality and maintainability, proving essential for extending the domain to incorporate new functionalities. Consequently, the modeling philosophy is "minimalism". Specifically, only base PDDL1.2 requirements such as `:strips` are used. While this occasionally results in more verbose formulations, it offers substantial benefits, including enhanced flexibility across different planners and compatibility with modeling paradigms like HDDL. Additionally, this approach forces the adoption of simpler domain representations, limiting how complex robot coordination can be expressed with a restricted set of operations. However, alternative modeling approaches are also explored in additional experiments.

A. Problem 1

The first problem is about base PDDL modelling of the environment, ignoring the presence of carriers, as they'll be added in sec. III-B. **Types.** The problem is formulated following the hierarchy at fig. 2, where `locatable` objects (including robots, boxes, supplies, patients, and medical units) can be positioned at specific `location` nodes, while

robot types are further specialized into worker_bot and helper_bot subtypes with distinct operational capabilities. **Predicates.** The domain organizes predicates into four main categories: spatial relationships (at for object locations, linked for facility connectivity), robot states (bot_is_free, carrying_box, helping_patient), object states (box_is_empty, box_on_ground, patient_on_ground, supply_in_box), and goal-oriented relationships that track medical unit needs (needs_supply, needs_patient) and their fulfillment (has_supply, has_patient). **Actions.** The following set of actions is defined:

- **move_bot_alone**: Enables any robot type to traverse between directly connected locations when not engaged in other tasks. The action requires the robot to be free (not carrying boxes or helping patients) and only permits movement along explicitly linked location pairs.
- **move_worker_bot_with_box**: Allows worker bots to transport boxes while moving between linked locations. The action is only available when the robot is actively carrying a box, and automatically updates both robot and box locations simultaneously.
- **move_helper_bot_with_patient**: Permits helper bots to escort patients between connected locations during assistance, updating both bot and patient's locations.
- **pick_up_box**: Enables worker bots to acquire boxes from the ground at their current location. The action requires co-location of robot and box, enforces mutual exclusion through the box_on_ground predicate, and transitions the robot from free to busy state. Only one box can be carried per robot at any time.
- **drop_down_box**: Allows worker bots to release carried boxes onto the ground at their current location. The action reverses the pickup operation by restoring the box_on_ground state and freeing the robot for other tasks. The box remains at the robot's current location after being dropped.
- **fill_box**: Loads supplies into empty boxes when robot, box, and supply are co-located. The action enforces a one-supply-per-box constraint through the box_is_empty precondition and requires the box to be on the ground rather than being carried. The modeling approach treats supplies as infinitely available resources at their designated locations (always central_warehouse in this case), with no inventory tracking or supply depletion mechanics implemented in the domain. A consequence of this design is that supply locations are never updated, as the only relevant information is where to retrieve them in order to perform a fill operation, or wether or not they're in a box in order to perform a delivery to medical units.
- **deliver_supply**: Transfers supplies from boxes to medical units that specifically require them. The action requires exact matching between the supply in the box and the medical unit's needs, automatically empties the box, and transitions the medical unit from needing to having the supply. The delivery must occur when box_on_ground.
- **pick_up_patient**: Initiates helper bot assistance for patients at shared locations. The action establishes an exclusive helping relationship through the helping_patient predicate, removes the patient from independent status (on_ground), and marks the helper bot as busy. Only one patient can be assisted per helper bot.
- **drop_down_patient**: Terminates helper bot assistance, leaving patients at the current location and freeing the helper bot for new assignments.
- **deliver_patient**: Transfers patients to medical units for treatment when the patient has been released from helper bot assistance. The action requires the patient to be in an independent status, ensures exact matching between the specific patient and medical unit requirements, and fulfills the medical unit's need by transitioning from needing to having the patient. Both patient and medical unit must be co-located for the delivery to occur.

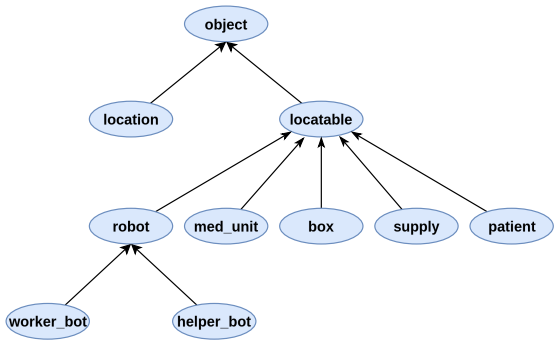

Fig. 2: Type hierarchy. Ellipses denote types, and arrows indicate the "extends" relationship.

B. Problem 2

The second problem extends the base PDDL modeling by introducing carriers to enable multi-box transportation capabilities. **Types.** The type hierarchy is extended with a new carrier type as a subtype of object, allowing robots to own and operate carriers with varying capacities. Carriers are treated as object extensions instead of locatable subtypes because carrier location is made implicit through bot ownership, moreover in this context there's no particular benefit in tracking their location. **Predicates.** The domain introduces carrier-specific predicates including ownership relationships (bot_owns_carrier), capacity constraints (carrier_can_1, carrier_can_2, carrier_can_3), occupancy states (carrier_empty, carrier_has_1, carrier_has_2, carrier_has_3), and box-carrier associations (carrying_box now relates carriers to boxes rather than robots to boxes). **Actions.** The following actions are introduced:

- **move_worker_bot_with_N_boxes**: Extends movement capabilities to handle multi-box transportation using carriers. The action is instantiated for different capacities (1, 2, or 3 boxes), requiring explicit carrier ownership and enforcing box distinctness through inequality constraints from `:equality`. Updates robot and all carried box locations simultaneously while maintaining carrier state consistency.
- **pick_up_Nth_box**: Enables incremental carrier loading across different occupancy states. The action transitions carriers between capacity levels (empty→1, 1→2, 2→3), enforcing capacity constraints and ownership verification. Robot availability status is updated only when transitioning from empty to occupied state.
- **drop_down_Nth_box**: Supports incremental carrier unloading across occupancy states. The action transitions carriers between capacity levels (3→2, 2→1, 1→empty), restoring box ground status and maintaining carrier occupancy tracking. Robot availability is restored only when transitioning to empty state.

The implementation assumes a maximum carrying capacity of 3 boxes but can be extended to higher capacities by iterating the same modeling philosophy with additional capacity predicates and corresponding action variants. Such approach avoids the need for numeric fluents, at the cost of defining manually an increasing number of near-identical actions and predicates.

C. Problem 3

Problem 3 involves extending the implementation from sec. III-B through HDDL modeling. This requires taking the previously defined primitive actions and designing appropriate tasks with their corresponding methods. **Tasks.** Two high-level tasks are identified: `deliver_patient_to_medical_unit` to orchestrate helper bot behavior for patient escort and delivery, and `deliver_supply_to_medical_unit` to manage worker bots with carriers for supply delivery. The complete task decomposition is illustrated in fig. 3. Both task designs explicitly encode through hierarchical decomposition the optimal planning outcomes described in sec. II-C, while providing flexibility and dynamic carrier utilization for worker bots.

- **deliver_patient_to_medical_unit**: The distinctive component is the recursive `route_helper_bot` subtask, which models helper bot movement (with or without patient) between arbitrary locations A and B (not necessarily adjacent). Beyond navigation, the design follows sec. II-C.
- **deliver_supply_to_medical_unit**: Movement is modeled recursively as in the helper bot case, with the primary design complexity focusing on optimal carrier management. For each supply requirement, the system includes an optional `retrieve_supply` sub-task that is executed in `deliver_supply_complete` and bypassed in `deliver_supply_skip_retrieval` methods. This structure enables two flexible execution paths, both ending with routing to the medical unit location and performing delivery. For each supply delivery task, the "complete" delivery method first ensures an empty box is available through the `handle_box_acquisition` task, which checks three possible scenarios: whether the robot is already carrying an empty box, whether an empty box exists at the supply location, or whether an empty box must be retrieved from elsewhere; after that, the sequence ground → fill → pickup is executed, with "ground" being an additional task that ensures the box is unloaded from the bot (if needed). After each ground, fill or pickup, the planner can optionally execute the relative "bulk" task, which is designed to optimize carrier utilization for future supply deliveries. Bulk tasks provide flexibility by offering: a "noop" method that skips the bulk optimization, or an active method that performs additional operations when beneficial. For instance, `pick_up_boxes_bulk` can either do nothing or pick up an arbitrary amount of additional boxes. To summarize, a worker bot prioritizes completing the required supply delivery while simultaneously using bulk operations to optimize the carrier for future deliveries. This optimization is achieved through the skip mechanism: when the bot already carries a box containing the target supply, it can bypass the entire retrieval process and proceed directly to delivery.

D. Problem 4

It's requested to reformulate sec. III-B into a temporal planning framework, by defining durative actions and allowing parallel execution wherever it's physically plausible. To achieve this scope, the domain extends the PDDL requirements to include `:durative-actions`, transforming all primitive actions into temporal variants with explicit duration specifications and proper mutex management. **Predicates.** The domain introduces a critical mutex predicate `bot_is_unlocked` to prevent concurrent action execution on the same robot, ensuring physical consistency while enabling parallelism across different robots. This predicate acts as a resource lock, acquired at action start and released at action end, preventing race conditions and conflicting state modifications. **Actions.** All actions from sec. III-B are reformulated as durative actions with the following key modifications: duration specification, temporal conditions, mutex management, and effect timing. Each action now has a fixed duration as shown in tab. I, reflecting realistic execution times. Preconditions are distributed across `at start`, `over all`, and `at end` timepoints, enabling realistic constraint checking throughout execution. All actions acquire exclusive robot access via `bot_is_unlocked` at the start and release it at the end, preventing concurrent execution on the same robot while allowing parallel operations across different robots. State changes are strategically placed at `at start` for immediate updates (e.g., location changes) or `at end` for completion-dependent effects (e.g., capacity updates).

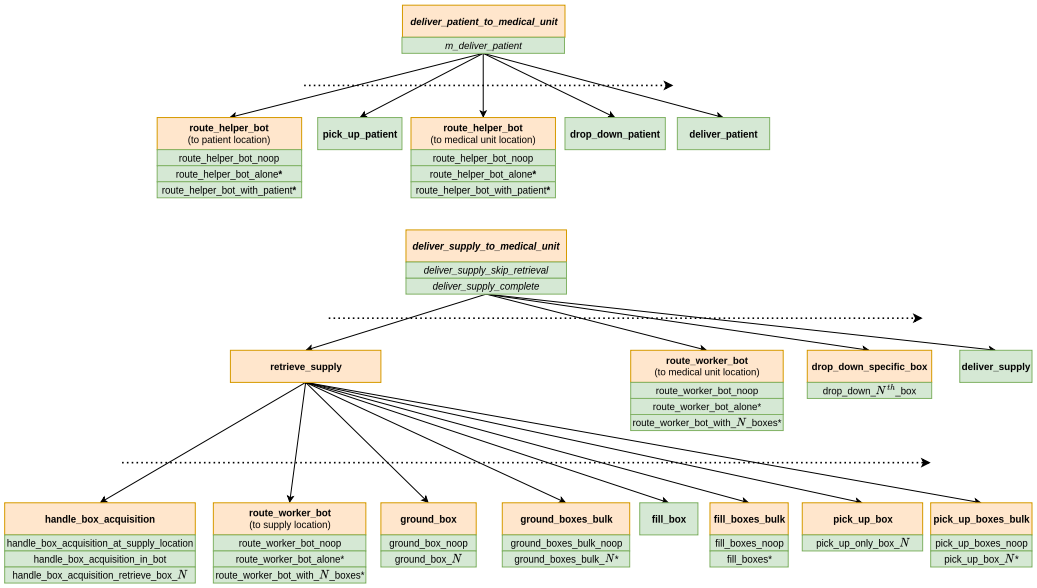

Fig. 3: High level Task decompositions. Orange rectangles represent tasks, green rectangles represents either primitives when connected to tasks by solid line arrows or methods when positioned directly below a task. Methods ending with * as name implement recursion; Methods containing N in the name denote the presence of multiple methods under the same name which are dependent or carrier capacity state; dotted arrows show ordering constraints.

Action	Duration
`move_bot_alone`	3
`move_worker_bot_with_N_boxes`	5
`move_helper_bot_patient`	5
`pick_up_Nth_box`	2
`drop_down_Nth_box`	2
`fill_box`	2
`deliver_supply`	1
`pick_up_patient`	2
`drop_down_patient`	2
`deliver_patient`	1

TABLE I: Durative action specifications with duration times. Durations are set to reflect the nature of each action; for example, a helper bot escorting a patient is assigned a longer duration to reflect slower movement.

E. Problem 5

This problem involves integrating the action implementations from sec. III-D into the PlanSys2 [4] toolbox, enabling their execution within the ROS2 system. This integration highlights the project's potential for real-world robotic applications. As per PlanSys2 guidelines, each action is modeled as a C++ class inheriting from `ActionExecutorClient`. These classes simulate action execution, meaning they mimic the behavior of a real-world operation without directly controlling physical hardware. The toolbox references an adapted version of the domain file from sec. III-D, ensuring correct handling of arguments, preconditions, effects, and durations for these simulated actions as well as allowing parallel execution of non-mutex tasks. This approach allows for the testing and validation of planning logic within a ROS2 environment without requiring actual robot hardware, effectively bridging the gap towards practical deployment.

IV. Results

The performance of the proposed solutions is evaluated within the planutils [5] framework, which provides a variety of planners suitable for all problems. Each problem is assessed across multiple scenarios of increasing complexity. Optimal solutions are also provided whenever computational constraints permit.

A. Problem 1

The benchmarking of problems sec. III-A and sec. III-B is given particular attention, as they form the foundation for sec. III-C, sec. III-D and sec. III-E. These benchmarks offer variety in task complexity:

1) `p1_0`: This problem involves one bot per type, one box, two supply deliveries, a single patient delivery, and the goal of restoring the initial state.
2) `p1_1`: Similar to `p1_0`, but with an additional worker bot and an extra box. The "restore initial state" goal is relaxed. This configuration allows for the examination of coordination among multiple bots and boxes.
3) `p1_2`: Based on `p1_0`, but includes one more delivery per robotic agent, and the goal is relaxed as in `p1_1`.
4) `p1_3`: This problem builds on `p1_1` but features two bots per type and an additional box, presenting a significant challenge.

Each problem is evaluated using three different planners: Fast-Downward [3], LAMA [7], and LAMA-First. The results are presented in Tables tab. II, tab. III, tab. IV, and tab. V.

As problem size increases, finding optimal solutions becomes extremely difficult due to an exponential increase of the search space; such behavior is made evident when comparing `p1_0` with `p1_1` and `p1_2`. For this reason, optimal solution search was omitted for `p1_3`. Nonetheless, sub-optimal planners like LAMA-First and LAMA generate good solutions within a reasonable execution time, making them practical for real-world applications. All problems were solved correctly, avoiding redundant trips on optimal planning output. Examples of planning outputs can be viewed at out. 1, out. 2, out. 3 and out. 4.

Planner	Heuristic	Time (s)	Steps	Optimal
Fast-Downward (A*)	LM-cut	4.25	33	✓
LAMA (2nd solution)	ff	10.55	33	✓
LAMA-First	ff	0.57	35	✗

TABLE II: `p1_0` results

Planner	Heuristic	Time (s)	Steps	Optimal
Fast-Downward (A*)	LM-cut	22.75	20	✓
LAMA (1st solution)	ff	0.004	20	✓
LAMA-First	ff	0.60	20	✓

TABLE III: `p1_1` results

Planner	Heuristic	Time (s)	Steps	Optimal
Fast-Downward (A*)	LM-cut	213.33	47	✓
LAMA (4th solution)	ff	0.34	47	✓
LAMA-First	ff	0.57	52	✗

TABLE IV: `p1_2` results

Planner	Heuristic	Time (s)	Steps	Optimal
Fast-Downward (A*)	LM-cut	$\gg$ 2782.10	-	-
LAMA (4th solution)	ff	0.74	43	✗
LAMA-First	ff	0.62	51	✗

TABLE V: `p1_3` results

B. Problem 2

Benchmarking follows the same approach as sec. IV-A, with increased domain complexity arising from the inclusion of carriers, which significantly alter the expected planning outcomes. From now onward, only two test problems versions will be presented per problem: one that meets the minimum requirements to evaluate solution correctness, and a second version designed specifically to test system limits. The evaluation covers:

1) `p2_0`: One bot of each type, with the worker bot equipped with a carrier having maximum capacity of 3, 2 boxes available, a total of 2 supply deliveries to complete, 1 patient to escort, and the objective of returning to the initial state.
2) `p2_1`: A challenging version of `p2_0` featuring 2 worker bots (one with carrier capacity of 3, the other with capacity of 2), 2 helper bots, 4 available boxes, 5 supply deliveries to complete, and 3 patients to escort. The requirement to restore the original state is removed.

Planner	Heuristic	Time (s)	Steps	Optimal
Fast-Downward (A*)	LM-cut	40.13	27	✓
LAMA (1st solution)	ff	0.007	27	✓
LAMA-First	ff	0.68	27	✓

TABLE VI: `p2_0` results

Planner	Heuristic	Time (s)	Steps	Optimal
LAMA (4th solution)	ff	118.50	60	✗
LAMA-First	ff	0.77	67	✗

TABLE VII: `p2_1` results

As demonstrated in out. 5, the planner successfully achieves optimal routing by leveraging the carrier mechanism, as well as respecting carrier capacity limits. This capability extends to more challenging problem variants featuring substantially larger delivery volumes. The results in out. 6 clearly illustrate that the two worker bots semi-efficiently distribute box loads among themselves and execute near-optimal paths to their designated medical units for delivery completion. **Domain Variants.** The aforementioned benchmark implements the foundations for subsequent problem formulations. To further experiment with PDDL modeling on this problem, two additional domain variations are presented:

1) **`numeric`**: A more compact and human-readable domain implementation utilizing `:numeric-fluents`. While this approach offers reduced compatibility across diverse planning systems, it substantially decreases verbosity through the incorporation of `when` and `forall` constructs, moreover, it makes much simpler the extension of carrier capacity over the pre-determined limit of 3 with very little effort, without requiring the implementation of new actions and predicates. Evaluation was conducted on a modified version of `p2_0` using the Expressive Numeric Heuristic Search Planner (ENHSP) [2], with findings documented in tab. VIII and generated output available at out. 7.
2) **`cost_sensitive`**: A domain identical to the baseline implementation, enriched with `:action-costs` and `total-cost` minimization as planning objective. The motivation behind this variation is the following hypothesis: assigning action costs inversely proportional to action convenience should make search better informed, enhancing planner performances. For example, to encourage carrier utilization, the first box loading operation has a cost of 5, the second box load costs 2, while the third costs 1. Simultaneously, bot movement costs are much larger if the bot is moving alone, and decrease the more boxes are being carried. All costs are specified at tab. X. The results presented in tab. IX validate this hypothesis for the tested domain, due to the notable improvements observed in the LAMA-First solution out. 8, which achieves optimal solution quality in a fraction of compute time.

Planner	Heuristic	Time (s)	Steps	Optimal
ENHSP (WAStar)	$W = 0.25$	71.59	27	✓

TABLE VIII: `p2_numeric` results

Planner	Heuristic	Time (s)	Steps	Optimal
Fast-Downward (A*)	LM-cut	56.0	27	✓
LAMA (3rd solution)	ff	0.32	27	✓
LAMA-First	ff	0.03	27	✓

TABLE IX: `p2_cost_sensitive` results

Action	Cost
`move_bot_alone`	10
`move_worker_bot_with_N_boxes`	5,2,1
`move_helper_bot_patient`	1
`pick_up_Nth_box`	5,2,1
`drop_down_Nth_box`	5,2,1
`fill_box`	1
`deliver_supply`	1
`pick_up_patient`	1
`drop_down_patient`	1
`deliver_patient`	1

TABLE X: Action costs for cost sensitive implementation of sec. III-B. Costs are inversely proportional to action convenience. When specified in the action name N denotes multiple action under the same name, specifically $N = \{1, 2, 3\}$ in this given order.

C. Problem 3

The proposed HDDL solution is evaluated using Planning and Acting in a Network Decomposition Architecture (PANDA) [6]. The test problem instances are:

- `p3_0`: A single worker bot with carrier capacity of 3 and 3 deliveries to complete, where 2 deliveries share the same destination location. The helper bot was excluded to simplify the task and focus on testing the discovery of "bulk" operations.
- `p3_1`: Based on `p3_0` but with the addition of 2 patient deliveries and one helper bot.

First, optimal solution reachability was tested by running PANDA directly from source code implementation. This approach yielded poor results, leading to the adoption of the planutils version instead. However, PANDA on planutils has a major limitation: it provides very few configuration options. Basic settings like continuing the search after finding the first solution or selecting different search methods cannot be adjusted. When attempting to use PANDA from source code, all trials failed due to memory constraints. This occurred even when using the fastest and most memory-efficient settings available. In contrast, running the exact same problem using planutils produced a working solution within seconds. This difference in performance is puzzling, but leads to the conclusion that the planutils PANDA implementation appears to be much more efficient. Unfortunately, this limitation impacts the variety of the benchmark results.

Results at tab. XI shows that the modelled domain allows for discovery of "bulk" operations (out. 9) shortcuts with good performances. When increasing the problem size, naturally it gets more complicated, specifically the used search is more likely to skip "bulk" operations as they provide a better local heuristics, results at tab. XII and out. 10

Planner	Heuristic	Time (s)	Steps	Optimal
PANDA (hhRC2)	ff;dist.;corr.count	0.1	22	✗

TABLE XI: `p3_0` results

Planner	Heuristic	Time (s)	Steps	Optimal
PANDA (hhRC2)	ff;dist.;corr.count	0.08	46	✗

TABLE XII: `p3_1` results

D. Problem 4

Temporal planning solutions are tested using the Optimizing Preferences and Time-dependent Costs (OPTIC) planner [1] on the following problem instances:

- `p4_0`: Identical to `p2_0`. The focus is on evaluating the correctness of the implemented temporal-mutex operator, which should permit different robots to operate simultaneously while preventing any single robot from performing multiple actions at the same time.
- `p4_1`: Identical to `p2_1`, providing a more demanding test case.

Both problems demonstrate expected behavior. As shown in out. 11 and out. 12, robots can operate in parallel while ensuring that each individual robot performs only one action at any given moment. Solutions are found within reasonable time limits and, while likely suboptimal, demonstrate good quality as shown in tab. XIII and tab. XIV.

Planner	Heuristic	Time (s)	Makespan	Optimal
OPTIC (WA*)	W=5	9.48	56.02	✗

TABLE XIII: `p4_0` results

Planner	Heuristic	Time (s)	Makespan	Optimal
OPTIC (WA*)	W=5	114.96	123.04	✗

TABLE XIV: `p4_1` results

E. Problem 5

Given that the implementation for this scenario closely mirrors the approach described in sec. III-D, it's proposed a single test case `p5_0` that replicates `p4_0` to assess the accuracy of domain translation within the `PlanSys2` framework. Truncated terminal outputs are presented in out. 13 and out. 14, demonstrating successful system execution. After `get plan`, a suboptimal plan is generated, followed by simulation initiation upon executing `run`, which displays progress indicators for each action. In practical deployments, these placeholder actions would be replaced with actual robot commands, such as sensor data acquisition routines or robotic kinematic computations. This particular task demonstrates the significant potential the tool used in this assignment could have on real-world applications.

V. Conclusions

This work successfully developed and validated a comprehensive healthcare planning system across multiple paradigms, demonstrating the practical application of planning in robotic logistics. The progressive implementation from basic PDDL through hierarchical decomposition, temporal planning, and ROS2 integration achieved complete functional coverage of healthcare facility coordination challenges.

The foundational PDDL implementation established robust domain modeling for multi-agent coordination, successfully handling supply delivery and patient escort scenarios. The carrier-enhanced transportation model effectively enabled multi-box optimization, reducing operational complexity through efficient bulk operations. Hierarchical HDDL modeling demonstrated effective task decomposition with recursive routing structures, while temporal planning achieved realistic parallel execution across multiple robotic agents with proper mutex constraints. The culminating ROS2 integration validated practical applicability, successfully executing complex plans within a robotic framework. Cost-sensitive domain variants showed significant performance improvements, confirming that well-designed action costs and domain knowledge injection can substantially enhance planning efficiency. The minimalist modeling philosophy proved advantageous for cross-paradigm compatibility, enabling seamless transitions between different planning formulations. Performance analysis confirmed the approach scales effectively with problem complexity, with various planning strategies providing practical solutions across different scenario sizes.

This systematic exploration demonstrates that planning techniques can successfully address complex healthcare logistics challenges, bridging theoretical planning research with practical robotic applications.

References

[1] J. Benton, Amanda Coles, and Andrew Coles. Temporal planning with preferences and time-dependent continuous costs. *ICAPS 2012 - Proceedings of the 22nd International Conference on Automated Planning and Scheduling*, 22, 05 2012.

[2] Scala Enrico, Haslum Patrik, Thiebaux Sylvie, and Ramirez Miquel. *Interval-Based Relaxation for General Numeric Planning*. 01 2016.

[3] Malte Helmert. The fast downward planning system. *CoRR*, abs/1109.6051, 2011.

[4] Francisco Martín, Jonatan Ginés, Vicente Matellán, and Francisco J. Rodríguez. Plansys2: A planning system framework for ros2, 2021.

[5] Christian Muise, Florian Pommerening, Jendrik Seipp, and Michael Katz. Planutils: Bringing planning to the masses. In *ICAPS 2022 System Demonstrations*, 2022.

[6] Institute of Artificial Intelligence at Ulm University. Panda: Planning and acting in a network decomposition architecture. https://panda-planner-dev.github.io/, 2018. Accessed: 2025-07-30.

[7] S. Richter and M. Westphal. The lama planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39:127–177, September 2010.

APPENDIX

```
(fill_box wbot1 box1 defibrillator central_warehouse)
(pick_up_patient hbot1 patient1 entrance)
(move_helper_bot_with_patient hbot1 patient1 entrance reception)
(move_helper_bot_with_patient hbot1 patient1 reception corridor1)
(move_helper_bot_with_patient hbot1 patient1 corridor1 surgical_block)
(drop_down_patient hbot1 patient1)
(deliver_patient hbot1 patient1 operation_unit surgical_block)
(move_bot_alone hbot1 surgical_block corridor1)
(move_bot_alone hbot1 corridor1 reception)
(move_bot_alone hbot1 reception entrance)
(pick_up_box wbot1 box1 central_warehouse)
(move_worker_bot_with_box wbot1 box1 central_warehouse corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 reception)
(move_worker_bot_with_box wbot1 box1 reception emergency_room)
(drop_down_box wbot1 box1)
(deliver_supply wbot1 box1 defibrillator emergency_room resuscitation_unit)
(pick_up_box wbot1 box1 emergency_room)
(move_worker_bot_with_box wbot1 box1 emergency_room reception)
(move_worker_bot_with_box wbot1 box1 reception corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 central_warehouse)
(drop_down_box wbot1 box1)
(fill_box wbot1 box1 scalpel central_warehouse)
(pick_up_box wbot1 box1 central_warehouse)
(move_worker_bot_with_box wbot1 box1 central_warehouse corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 reception)
(move_worker_bot_with_box wbot1 box1 reception emergency_room)
(drop_down_box wbot1 box1)
(deliver_supply wbot1 box1 scalpel emergency_room trauma_unit)
(pick_up_box wbot1 box1 emergency_room)
(move_worker_bot_with_box wbot1 box1 emergency_room reception)
(move_worker_bot_with_box wbot1 box1 reception corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 central_warehouse)
(drop_down_box wbot1 box1)
; cost = 33 (unit cost)
```

Listing 1: `p1_0` Fast-Downward output

```
(fill_box wbot1 box2 defibrillator central_warehouse)
(pick_up_patient hbot1 patient1 entrance)
(move_helper_bot_with_patient hbot1 patient1 entrance reception)
(pick_up_box wbot2 box2 central_warehouse)
(fill_box wbot1 box1 scalpel central_warehouse)
(pick_up_box wbot1 box1 central_warehouse)
(move_worker_bot_with_box wbot1 box1 central_warehouse corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 reception)
(move_worker_bot_with_box wbot1 box1 reception emergency_room)
(drop_down_box wbot1 box1)
(deliver_supply wbot1 box1 scalpel emergency_room trauma_unit)
(move_worker_bot_with_box wbot2 box2 central_warehouse corridor1)
(move_worker_bot_with_box wbot2 box2 corridor1 reception)
(move_worker_bot_with_box wbot2 box2 reception emergency_room)
(drop_down_box wbot2 box2)
(deliver_supply wbot1 box2 defibrillator emergency_room resuscitation_unit)
(move_helper_bot_with_patient hbot1 patient1 reception corridor1)
(move_helper_bot_with_patient hbot1 patient1 corridor1 surgical_block)
(drop_down_patient hbot1 patient1)
(deliver_patient hbot1 patient1 operation_unit surgical_block)
; cost = 20 (unit cost)
```

Listing 2: `p1_1` Fast-Downward output

```
(fill_box wbot1 box1 anesthetic central_warehouse)
(pick_up_box wbot1 box1 central_warehouse)
(move_worker_bot_with_box wbot1 box1 central_warehouse corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 diagnostic_lab)
(move_worker_bot_with_box wbot1 box1 diagnostic_lab neurology_ward)
(drop_down_box wbot1 box1)
(deliver_supply wbot1 box1 anesthetic neurology_ward neurology_unit)
(pick_up_box wbot1 box1 neurology_ward)
(move_worker_bot_with_box wbot1 box1 neurology_ward diagnostic_lab)
(move_worker_bot_with_box wbot1 box1 diagnostic_lab corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 central_warehouse)
(drop_down_box wbot1 box1)
(fill_box wbot1 box1 defibrillator central_warehouse)
(pick_up_box wbot1 box1 central_warehouse)
(move_worker_bot_with_box wbot1 box1 central_warehouse corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 reception)
(move_worker_bot_with_box wbot1 box1 reception emergency_room)
(drop_down_box wbot1 box1)
(deliver_supply wbot1 box1 defibrillator emergency_room resuscitation_unit)
(pick_up_box wbot1 box1 emergency_room)
(move_worker_bot_with_box wbot1 box1 emergency_room reception)
(move_worker_bot_with_box wbot1 box1 reception corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 central_warehouse)
(drop_down_box wbot1 box1)
(fill_box wbot1 box1 scalpel central_warehouse)
(pick_up_box wbot1 box1 central_warehouse)
(move_worker_bot_with_box wbot1 box1 central_warehouse corridor1)
(move_worker_bot_with_box wbot1 box1 corridor1 reception)
(move_worker_bot_with_box wbot1 box1 reception emergency_room)
(drop_down_box wbot1 box1)
(deliver_supply wbot1 box1 scalpel emergency_room trauma_unit)
(pick_up_patient hbot1 patient1 entrance)
(move_helper_bot_with_patient hbot1 patient1 entrance reception)
(move_helper_bot_with_patient hbot1 patient1 reception corridor1)
(move_helper_bot_with_patient hbot1 patient1 corridor1 surgical_block)
(drop_down_patient hbot1 patient1)
(deliver_patient hbot1 patient1 operation_unit surgical_block)
(move_bot_alone hbot1 surgical_block corridor1)
(move_bot_alone hbot1 corridor1 reception)
(move_bot_alone hbot1 reception entrance)
(pick_up_patient hbot1 patient2 entrance)
(move_helper_bot_with_patient hbot1 patient2 entrance reception)
(move_helper_bot_with_patient hbot1 patient2 reception corridor1)
(move_helper_bot_with_patient hbot1 patient2 corridor1 surgical_block)
(move_helper_bot_with_patient hbot1 patient2 surgical_block cardiology_ward)
(drop_down_patient hbot1 patient2)
(deliver_patient hbot1 patient2 cardiology_unit cardiology_ward)
; cost = 47 (unit cost)
```

Listing 3: `p1_2` Fast-Downward output

```
(pick_up_patient hbot1 patient2 entrance)
(move_helper_bot_with_patient hbot1 patient2 entrance reception)
(move_helper_bot_with_patient hbot1 patient2 reception corridor1)
(move_helper_bot_with_patient hbot1 patient2 corridor1 surgical_block)
(move_helper_bot_with_patient hbot1 patient2 surgical_block cardiology_ward)
(fill_box wbot1 box2 anesthetic central_warehouse)
(drop_down_patient hbot1 patient2)
(pick_up_box wbot1 box2 central_warehouse)
(deliver_patient hbot1 patient2 cardiology_unit cardiology_ward)
(fill_box wbot1 box1 defibrillator central_warehouse)
(move_worker_bot_with_box wbot1 box2 central_warehouse corridor1)
(move_worker_bot_with_box wbot1 box2 corridor1 diagnostic_lab)
(move_worker_bot_with_box wbot1 box2 diagnostic_lab neurology_ward)
(move_bot_alone hbot1 cardiology_ward surgical_block)
(drop_down_box wbot1 box2)
(deliver_supply wbot1 box2 anesthetic neurology_ward neurology_unit)
(pick_up_box wbot1 box2 neurology_ward)
(move_worker_bot_with_box wbot1 box2 neurology_ward diagnostic_lab)
(move_worker_bot_with_box wbot1 box2 diagnostic_lab corridor1)
(move_bot_alone hbot2 reception entrance)
(pick_up_patient hbot2 patient1 entrance)
(move_helper_bot_with_patient hbot2 patient1 entrance reception)
(move_helper_bot_with_patient hbot2 patient1 reception corridor1)
(move_helper_bot_with_patient hbot2 patient1 corridor1 surgical_block)
(drop_down_patient hbot2 patient1)
(deliver_patient hbot1 patient1 operation_unit surgical_block)
(move_worker_bot_with_box wbot1 box2 corridor1 central_warehouse)
(drop_down_box wbot1 box2)
(fill_box wbot1 box2 scalpel central_warehouse)
(pick_up_box wbot1 box2 central_warehouse)
(move_worker_bot_with_box wbot1 box2 central_warehouse corridor1)
(move_worker_bot_with_box wbot1 box2 corridor1 reception)
(move_worker_bot_with_box wbot1 box2 reception emergency_room)
(move_bot_alone wbot2 surgical_block corridor1)
(drop_down_box wbot1 box2)
(deliver_supply wbot1 box2 scalpel emergency_room trauma_unit)
(move_bot_alone wbot2 corridor1 central_warehouse)
(pick_up_box wbot2 box1 central_warehouse)
(move_worker_bot_with_box wbot2 box1 central_warehouse corridor1)
(move_worker_bot_with_box wbot2 box1 corridor1 reception)
(move_worker_bot_with_box wbot2 box1 reception emergency_room)
(drop_down_box wbot2 box1)
(deliver_supply wbot1 box1 defibrillator emergency_room resuscitation_unit)
; cost = 43 (unit cost)
```

Listing 4: `p1_3` LAMA (4^{th} solution) output

```
(fill_box wbot1 box1 defibrillator central_warehouse)
(fill_box wbot1 box2 scalpel central_warehouse)
(pick_up_1st_box wbot1 carrier1 box1 central_warehouse)
(pick_up_2nd_box wbot1 carrier1 box2 central_warehouse)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 central_warehouse corridor1)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 corridor1 reception)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 reception emergency_room)
(drop_down_2nd_box wbot1 carrier1 box1)
(drop_down_1st_box wbot1 carrier1 box2)
(pick_up_patient hbot1 patient1 entrance)
(deliver_supply wbot1 box1 defibrillator emergency_room resuscitation_unit)
(deliver_supply wbot1 box2 scalpel emergency_room trauma_unit)
(move_helper_bot_patient hbot1 entrance reception patient1)
(move_helper_bot_patient hbot1 reception corridor1 patient1)
(move_helper_bot_patient hbot1 corridor1 surgical_block patient1)
(drop_down_patient hbot1 patient1)
(deliver_patient hbot1 patient1 operation_unit surgical_block)
(move_bot_alone hbot1 surgical_block corridor1)
(move_bot_alone hbot1 corridor1 reception)
(move_bot_alone hbot1 reception entrance)
(pick_up_1st_box wbot1 carrier1 box1 emergency_room)
(pick_up_2nd_box wbot1 carrier1 box2 emergency_room)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 emergency_room reception)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 reception corridor1)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 corridor1 central_warehouse)
(drop_down_2nd_box wbot1 carrier1 box1)
(drop_down_1st_box wbot1 carrier1 box2)
; cost = 27 (unit cost)
```

Listing 5: `p2_0` Fast-Downward output

```
(fill_box wbot1 box4 blood_bag central_warehouse)
(fill_box wbot1 box3 defibrillator central_warehouse)
(fill_box wbot1 box2 scalpel central_warehouse)
(fill_box wbot1 box1 surgical_mask central_warehouse)
(move_bot_alone hbot1 entrance reception)
(move_bot_alone hbot1 reception corridor1)
(move_bot_alone hbot1 corridor1 diagnostic_lab)
(pick_up_patient hbot1 patient3 diagnostic_lab)
(move_helper_bot_patient hbot1 diagnostic_lab neurology_ward patient3)
(drop_down_patient hbot1 patient3)
(deliver_patient hbot1 patient3 neurology_unit neurology_ward)
(pick_up_patient hbot2 patient1 entrance)
(move_helper_bot_patient hbot2 entrance reception patient1)
```

```
(move_helper_bot_patient hbot2 reception corridor1 patient1)
(move_helper_bot_patient hbot2 corridor1 surgical_block patient1)
(drop_down_patient hbot2 patient1)
(deliver_patient hbot2 patient1 operation_unit surgical_block)
(pick_up_1st_box wbot1 carrier1 box1 central_warehouse)
(pick_up_1st_box wbot2 carrier2 box4 central_warehouse)
(move_worker_bot_with_1_box wbot2 carrier2 box4 central_warehouse corridor1)
(move_worker_bot_with_1_box wbot2 carrier2 box4 corridor1 diagnostic_lab)
(move_worker_bot_with_1_box wbot2 carrier2 box4 diagnostic_lab neurology_ward)
(drop_down_1st_box wbot2 carrier2 box4)
(pick_up_2nd_box wbot1 carrier1 box2 central_warehouse)
(pick_up_3rd_box wbot1 carrier1 box3 central_warehouse)
(move_bot_alone hbot1 neurology_ward diagnostic_lab)
(move_bot_alone hbot2 surgical_block cardiology_ward)
(move_bot_alone hbot1 diagnostic_lab corridor1)
(move_bot_alone hbot1 corridor1 reception)
(move_bot_alone hbot1 reception entrance)
(pick_up_patient hbot1 patient2 entrance)
(move_helper_bot_patient hbot1 entrance reception patient2)
(move_helper_bot_patient hbot1 reception corridor1 patient2)
(move_helper_bot_patient hbot1 corridor1 surgical_block patient2)
(move_helper_bot_patient hbot1 surgical_block cardiology_ward patient2)
(drop_down_patient hbot1 patient2)
(deliver_patient hbot1 patient2 cardiology_unit cardiology_ward)
(move_worker_bot_with_3_boxes wbot1 carrier1 box1 box2 box3 central_warehouse
      corridor1)
(deliver_supply wbot2 box4 blood_bag neurology_ward neurology_unit)
(move_worker_bot_with_3_boxes wbot1 carrier1 box1 box2 box3 corridor1 reception)
(drop_down_3rd_box wbot1 carrier1 box1)
(deliver_supply wbot1 box1 surgical_mask reception triage_unit)
(pick_up_3rd_box wbot1 carrier1 box1 reception)
(move_worker_bot_with_3_boxes wbot1 carrier1 box1 box2 box3 reception emergency_room
      )
(drop_down_3rd_box wbot1 carrier1 box2)
(drop_down_2nd_box wbot1 carrier1 box3)
(pick_up_1st_box wbot2 carrier2 box4 neurology_ward)
(move_worker_bot_with_1_box wbot2 carrier2 box4 neurology_ward diagnostic_lab)
(deliver_supply wbot1 box2 scalpel emergency_room trauma_unit)
(deliver_supply wbot1 box3 defibrillator emergency_room resuscitation_unit)
(move_worker_bot_with_1_box wbot2 carrier2 box4 diagnostic_lab corridor1)
(move_worker_bot_with_1_box wbot2 carrier2 box4 corridor1 central_warehouse)
(drop_down_1st_box wbot2 carrier2 box4)
(fill_box wbot2 box4 stethoscope central_warehouse)
(pick_up_1st_box wbot2 carrier2 box4 central_warehouse)
(move_worker_bot_with_1_box wbot2 carrier2 box4 central_warehouse corridor1)
(move_worker_bot_with_1_box wbot2 carrier2 box4 corridor1 surgical_block)
(move_worker_bot_with_1_box wbot2 carrier2 box4 surgical_block cardiology_ward)
(drop_down_1st_box wbot2 carrier2 box4)
(deliver_supply wbot2 box4 stethoscope cardiology_ward cardiology_unit)
; cost = 60 (unit cost)
```

Listing 6: `p2_1` LAMA (4^{th} solution) output

```
(fill_box wBot1 box1 defibrillator central_warehouse)
(pick_up_box wBot1 carrier1 box1 central_warehouse)
(fill_box wBot1 box2 scalpel central_warehouse)
(pick_up_box wBot1 carrier1 box2 central_warehouse)
(move_worker_bot_with_boxes wBot1 carrier1 central_warehouse corridor1)
(move_worker_bot_with_boxes wBot1 carrier1 corridor1 reception)
(move_worker_bot_with_boxes wBot1 carrier1 reception emergency_room)
(drop_down_box wBot1 carrier1 box2)
(drop_down_box wBot1 carrier1 box1)
(deliver_supply wBot1 box1 defibrillator emergency_room resuscitation_unit)
(deliver_supply wBot1 box2 scalpel emergency_room trauma_unit)
(pick_up_box wBot1 carrier1 box1 emergency_room)
(pick_up_box wBot1 carrier1 box2 emergency_room)
(move_worker_bot_with_boxes wBot1 carrier1 emergency_room reception)
(move_worker_bot_with_boxes wBot1 carrier1 reception corridor1)
(move_worker_bot_with_boxes wBot1 carrier1 corridor1 central_warehouse)
(drop_down_box wBot1 carrier1 box2)
(drop_down_box wBot1 carrier1 box1)
(pick_up_patient hBot1 patient1 entrance)
(move_helper_bot_patient hBot1 entrance reception patient1)
(move_helper_bot_patient hBot1 reception corridor1 patient1)
(move_helper_bot_patient hBot1 corridor1 surgical_block patient1)
(drop_down_patient hBot1 patient1)
(deliver_patient hBot1 patient1 operation_unit surgical_block)
(move_bot_alone hBot1 surgical_block corridor1)
(move_bot_alone hBot1 corridor1 reception)
(move_bot_alone hBot1 reception entrance)
```

Listing 7: `p2_numeric` ENHSP output

```
(pick_up_patient hbot1 patient1 entrance)
(move_helper_bot_patient hbot1 entrance reception patient1)
(move_helper_bot_patient hbot1 reception corridor1 patient1)
(move_helper_bot_patient hbot1 corridor1 surgical_block patient1)
(drop_down_patient hbot1 patient1)
(deliver_patient hbot1 patient1 operation_unit surgical_block)
(move_bot_alone hbot1 surgical_block corridor1)
(move_bot_alone hbot1 corridor1 reception)
(move_bot_alone hbot1 reception entrance)
(fill_box wbot1 box2 defibrillator central_warehouse)
(fill_box wbot1 box1 scalpel central_warehouse)
(pick_up_1st_box wbot1 carrier1 box1 central_warehouse)
(pick_up_2nd_box wbot1 carrier1 box2 central_warehouse)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 central_warehouse corridor1)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 corridor1 reception)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 reception emergency_room)
(drop_down_2nd_box wbot1 carrier1 box1)
(deliver_supply wbot1 box1 scalpel emergency_room trauma_unit)
(drop_down_1st_box wbot1 carrier1 box2)
(deliver_supply wbot1 box2 defibrillator emergency_room resuscitation_unit)
(pick_up_1st_box wbot1 carrier1 box1 emergency_room)
(pick_up_2nd_box wbot1 carrier1 box2 emergency_room)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 emergency_room reception)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 reception corridor1)
(move_worker_bot_with_2_boxes wbot1 carrier1 box1 box2 corridor1 central_warehouse)
(drop_down_2nd_box wbot1 carrier1 box1)
(drop_down_1st_box wbot1 carrier1 box2)
; cost = 80 (general cost)
```

Listing 8: `p2_cost_sensitive` LAMA-First output

```
==>
348 fill_box wBot1 box1 scalpel central_warehouse
830 fill_box wBot1 box3 bandages central_warehouse
864 fill_box wBot1 box2 defibrillator central_warehouse
875 pick_up_1st_box wBot1 carrier1 box1 central_warehouse
885 pick_up_2nd_box wBot1 carrier1 box2 central_warehouse
905 pick_up_3rd_box wBot1 carrier1 box3 central_warehouse
953 move_worker_bot_with_3_boxes wBot1 carrier1 box3 box1 box2 central_warehouse
      corridor1
1101 move_worker_bot_with_3_boxes wBot1 carrier1 box3 box1 box2 corridor1 reception
1439 move_worker_bot_with_3_boxes wBot1 carrier1 box3 box2 box1 reception
      emergency_room
1752 move_worker_bot_with_3_boxes wBot1 carrier1 box3 box2 box1 emergency_room
      reception
4506 move_worker_bot_with_3_boxes wBot1 carrier1 box3 box2 box1 reception
      emergency_room
4517 drop_down_3rd_box wBot1 carrier1 box2
5316 deliver_supply wBot1 box2 defibrillator emergency_room resuscitation_unit
5332 move_worker_bot_with_2_boxes wBot1 carrier1 box3 box1 emergency_room reception
5585 move_worker_bot_with_2_boxes wBot1 carrier1 box3 box1 reception emergency_room
5614 drop_down_2nd_box wBot1 carrier1 box1
5741 deliver_supply wBot1 box1 scalpel emergency_room trauma_unit
5742 move_worker_bot_with_1_box wBot1 carrier1 box3 emergency_room reception
5758 move_worker_bot_with_1_box wBot1 carrier1 box3 reception corridor1
5858 move_worker_bot_with_1_box wBot1 carrier1 box3 corridor1 surgical_block
5905 drop_down_1st_box wBot1 carrier1 box3
5906 deliver_supply wBot1 box3 bandages surgical_block operation_unit
root 1 13 347
1 deliver_supply_to_medical_unit bandages operation_unit ->
      m_deliver_supply_skip_retrieval 924 5903 5906
13 deliver_supply_to_medical_unit scalpel trauma_unit -> m_deliver_supply_complete
      5908 914 1736 5741
136 handle_box_acquisition wBot1 carrier1 box1 scalpel central_warehouse ->
      m_handle_box_acquisition_at_supply_location
328 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_noop
335 ground_box wBot1 carrier1 box1 -> ground_box_noop
340 ground_boxes_bulk wBot1 carrier1 -> m_ground_boxes_bulk_noop
347 deliver_supply_to_medical_unit defibrillator resuscitation_unit ->
      m_deliver_supply_skip_retrieval 892 1601 5316
374 fill_boxes_bulk wBot1 central_warehouse -> m_fill_box_bulk 830 854
854 fill_boxes_bulk wBot1 central_warehouse -> m_fill_box_bulk 864 870
870 fill_boxes_bulk wBot1 central_warehouse -> m_fill_boxes_bulk_noop
873 pick_up_box wBot1 carrier1 box1 central_warehouse -> m_pick_up_only_box_1 875
881 pick_up_boxes_bulk wBot1 carrier1 central_warehouse -> m_pick_up_box_2 885 902
892 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_noop
902 pick_up_boxes_bulk wBot1 carrier1 central_warehouse -> m_pick_up_box_3 905 909
909 pick_up_boxes_bulk wBot1 carrier1 central_warehouse -> m_pick_up_boxes_noop
914 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_noop
924 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_3_boxes 953
      1062
1062 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_3_boxes 1101
      1236
1236 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_3_boxes 1439
      1591
1591 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_3_boxes 1752
      1770
1601 drop_down_box wBot1 carrier1 box2 -> m_drop_down_3rd_box 4517
1736 drop_down_box wBot1 carrier1 box1 -> m_drop_down_2nd_box 5614
1770 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_3_boxes 4506
      4516
4516 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_2_boxes 5332
      5340
5340 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_2_boxes 5585
      5613
5613 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_1_box 5742
      5746
5746 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_1_box 5758
      5818
5818 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_1_box 5858
      5896
5896 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_noop
5903 drop_down_box wBot1 carrier1 box3 -> m_drop_down_1st_box 5905
5908 retrieve_supply wBot1 carrier1 box1 scalpel central_warehouse ->
      m_retrieve_supply 136 328 335 340 348 374 873 881
```

Listing 9: `p3_0` PANDA output

```
==>
216 fill_box wBot1 box1 scalpel central_warehouse
225 pick_up_1st_box wBot1 carrier1 box1 central_warehouse
522 pick_up_2nd_box wBot1 carrier1 box2 central_warehouse
2352 move_worker_bot_with_2_boxes wBot1 carrier1 box1 box2 central_warehouse
      corridor1
2480 move_worker_bot_with_2_boxes wBot1 carrier1 box1 box2 corridor1 reception
2680 move_worker_bot_with_2_boxes wBot1 carrier1 box2 box1 reception emergency_room
2742 drop_down_2nd_box wBot1 carrier1 box1
2743 deliver_supply wBot1 box1 scalpel emergency_room trauma_unit
3524 move_worker_bot_with_1_box wBot1 carrier1 box2 emergency_room reception
3612 move_worker_bot_with_1_box wBot1 carrier1 box2 reception corridor1
3712 move_worker_bot_with_1_box wBot1 carrier1 box2 corridor1 central_warehouse
3745 drop_down_1st_box wBot1 carrier1 box2
3750 fill_box wBot1 box2 defibrillator central_warehouse
3759 pick_up_1st_box wBot1 carrier1 box2 central_warehouse
3830 move_worker_bot_with_1_box wBot1 carrier1 box2 central_warehouse corridor1
3895 move_worker_bot_with_1_box wBot1 carrier1 box2 corridor1 reception
3918 move_worker_bot_with_1_box wBot1 carrier1 box2 reception emergency_room
```

```
3996 drop_down_1st_box wBot1 carrier1 box2
3997 deliver_supply wBot1 box2 defibrillator emergency_room resuscitation_unit
4353 move_bot_alone wBot1 emergency_room reception
4384 move_bot_alone wBot1 reception corridor1
4427 move_bot_alone wBot1 corridor1 central_warehouse
4461 fill_box wBot1 box3 bandages central_warehouse
4470 pick_up_1st_box wBot1 carrier1 box3 central_warehouse
4494 move_worker_bot_with_1_box wBot1 carrier1 box3 central_warehouse corridor1
4564 move_worker_bot_with_1_box wBot1 carrier1 box3 corridor1 surgical_block
4629 drop_down_1st_box wBot1 carrier1 box3
4630 deliver_supply wBot1 box3 bandages surgical_block operation_unit
4656 pick_up_patient hBot1 patient1 entrance
4730 move_helper_bot_patient hBot1 entrance reception patient1
4798 move_helper_bot_patient hBot1 reception corridor1 patient1
4830 move_helper_bot_patient hBot1 corridor1 surgical_block patient1
4901 move_helper_bot_patient hBot1 surgical_block cardiology_ward patient1
4924 drop_down_patient hBot1 patient1
4925 deliver_patient hBot1 patient1 cardiology_unit cardiology_ward
4973 move_bot_alone hBot1 cardiology_ward surgical_block
5002 move_bot_alone hBot1 surgical_block corridor1
5030 move_bot_alone hBot1 corridor1 reception
5298 move_bot_alone hBot1 reception entrance
5304 pick_up_patient hBot1 patient2 entrance
5328 move_helper_bot_patient hBot1 entrance reception patient2
5393 move_helper_bot_patient hBot1 reception corridor1 patient2
5423 move_helper_bot_patient hBot1 corridor1 diagnostic_lab patient2
5490 move_helper_bot_patient hBot1 diagnostic_lab neurology_ward patient2
5514 drop_down_patient hBot1 patient2
5515 deliver_patient hBot1 patient2 neurology_unit neurology_ward
root 8 2748 4005 5516 5520
8 deliver_supply_to_medical_unit scalpel trauma_unit -> m_deliver_supply_complete
      5517 1231 2740 2743
10 handle_box_acquisition wBot1 carrier1 box1 scalpel central_warehouse ->
      m_handle_box_acquisition_at_supply_location
202 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_noop
209 ground_box wBot1 carrier1 box1 -> ground_box_noop
214 ground_boxes_bulk wBot1 carrier1 -> m_ground_boxes_bulk_noop
220 fill_boxes_bulk wBot1 central_warehouse -> m_fill_boxes_bulk_noop
223 pick_up_box wBot1 carrier1 box1 central_warehouse -> m_pick_up_only_box_1 225
231 pick_up_boxes_bulk wBot1 carrier1 central_warehouse -> m_pick_up_box_2 522 1223
1223 pick_up_boxes_bulk wBot1 carrier1 central_warehouse -> m_pick_up_boxes_noop
1231 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_with_2_boxes 2352
      2449
2449 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_with_2_boxes 2480
      2597
2597 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_with_2_boxes 2680
      2732
2732 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_noop
2740 drop_down_box wBot1 carrier1 box1 -> m_drop_down_2nd_box 2742
2748 deliver_supply_to_medical_unit defibrillator resuscitation_unit ->
      m_deliver_supply_complete 5518 3769 3994 3997
3292 handle_box_acquisition wBot1 carrier1 box2 defibrillator central_warehouse ->
      m_handle_box_acquisition_in_bot
3485 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_with_1_box 3524
      3561
3561 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_with_1_box 3612
      3645
3645 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_with_1_box 3712
      3735
3735 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_noop
3743 ground_box wBot1 carrier1 box2 -> m_ground_box_1 3745
3748 ground_boxes_bulk wBot1 carrier1 -> m_ground_boxes_bulk_noop
3754 fill_boxes_bulk wBot1 central_warehouse -> m_fill_boxes_bulk_noop
3757 pick_up_box wBot1 carrier1 box2 central_warehouse -> m_pick_up_only_box_1 3759
3762 pick_up_boxes_bulk wBot1 carrier1 central_warehouse -> m_pick_up_boxes_noop
3769 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_with_1_box 3830
      3837
3837 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_with_1_box 3895
      3911
3911 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_with_1_box 3918
      3987
3987 route_worker_bot wBot1 emergency_room -> m_route_worker_bot_noop
3994 drop_down_box wBot1 carrier1 box2 -> m_drop_down_1st_box 3996
4005 deliver_supply_to_medical_unit bandages operation_unit ->
      m_deliver_supply_complete 5519 4480 4627 4630
4141 handle_box_acquisition wBot1 carrier1 box3 bandages central_warehouse ->
      m_handle_box_acquisition_at_supply_location
4334 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_alone 4353 4364
4364 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_alone 4384 4402
4402 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_alone 4427 4447
4447 route_worker_bot wBot1 central_warehouse -> m_route_worker_bot_noop
4454 ground_box wBot1 carrier1 box3 -> ground_box_noop
4459 ground_boxes_bulk wBot1 carrier1 -> m_ground_boxes_bulk_noop
4465 fill_boxes_bulk wBot1 central_warehouse -> m_fill_boxes_bulk_noop
4468 pick_up_box wBot1 carrier1 box3 central_warehouse -> m_pick_up_only_box_1 4470
4473 pick_up_boxes_bulk wBot1 carrier1 central_warehouse -> m_pick_up_boxes_noop
4480 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_1_box 4494
      4548
4548 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_with_1_box 4564
      4620
4620 route_worker_bot wBot1 surgical_block -> m_route_worker_bot_noop
4627 drop_down_box wBot1 carrier1 box3 -> m_drop_down_1st_box 4629
4654 route_helper_bot hBot1 entrance -> m_route_helper_bot_noop
4665 route_helper_bot hBot1 cardiology_ward -> m_route_helper_bot_with_patient 4730
      4739
4739 route_helper_bot hBot1 cardiology_ward -> m_route_helper_bot_with_patient 4798
      4821
4821 route_helper_bot hBot1 cardiology_ward -> m_route_helper_bot_with_patient 4830
      4873
4873 route_helper_bot hBot1 cardiology_ward -> m_route_helper_bot_with_patient 4901
      4922
4922 route_helper_bot hBot1 cardiology_ward -> m_route_helper_bot_noop
4951 route_helper_bot hBot1 entrance -> m_route_helper_bot_alone 4973 4979
4979 route_helper_bot hBot1 entrance -> m_route_helper_bot_alone 5002 5009
5009 route_helper_bot hBot1 entrance -> m_route_helper_bot_alone 5030 5049
5049 route_helper_bot hBot1 entrance -> m_route_helper_bot_alone 5298 5302
5302 route_helper_bot hBot1 entrance -> m_route_helper_bot_noop
5313 route_helper_bot hBot1 neurology_ward -> m_route_helper_bot_with_patient 5328
      5359
5359 route_helper_bot hBot1 neurology_ward -> m_route_helper_bot_with_patient 5393
      5409
5409 route_helper_bot hBot1 neurology_ward -> m_route_helper_bot_with_patient 5423
      5465
5465 route_helper_bot hBot1 neurology_ward -> m_route_helper_bot_with_patient 5490
      5512
5512 route_helper_bot hBot1 neurology_ward -> m_route_helper_bot_noop
5516 deliver_patient_to_medical_unit patient1 cardiology_unit -> m_deliver_patient
      4654 4656 4665 4924 4925
5517 retrieve_supply wBot1 carrier1 box1 scalpel central_warehouse ->
      m_retrieve_supply 10 202 209 214 216 220 223 231
5518 retrieve_supply wBot1 carrier1 box2 defibrillator central_warehouse ->
      m_retrieve_supply 3292 3485 3743 3748 3750 3754 3757 3762
5519 retrieve_supply wBot1 carrier1 box3 bandages central_warehouse ->
      m_retrieve_supply 4141 4334 4454 4459 4461 4465 4468 4473
5520 deliver_patient_to_medical_unit patient2 neurology_unit -> m_deliver_patient
      4951 5304 5313 5514 5515
```

Listing 10: `p3_1` PANDA output

```
; Plan found with metric 56.019
; States evaluated so far: 7724
; States pruned based on pre-heuristic cost lower bound: 116
; Time 9.48
0.000: (pick_up_1st_box wbot1 carrier1 box1 central_warehouse) [2.000]
0.000: (pick_up_patient hbot1 patient1 entrance) [2.000]
2.001: (move_helper_bot_patient hbot1 entrance reception patient1) [5.000]
2.001: (fill_box wbot1 box2 scalpel central_warehouse) [2.000]
4.002: (pick_up_2nd_box wbot1 carrier1 box2 central_warehouse) [2.000]
6.003: (drop_down_2nd_box wbot1 carrier1 box1) [2.000]
7.002: (move_helper_bot_patient hbot1 reception corridor1 patient1) [5.000]
8.004: (fill_box wbot1 box1 defibrillator central_warehouse) [2.000]
10.005: (pick_up_2nd_box wbot1 carrier1 box1 central_warehouse) [2.000]
12.003: (move_helper_bot_patient hbot1 corridor1 surgical_block patient1) [5.000]
12.006: (move_worker_bot_with_2_boxes wbot1 carrier1 box2 box1 central_warehouse
      corridor1) [5.000]
17.004: (drop_down_patient hbot1 patient1) [2.000]
17.007: (move_worker_bot_with_2_boxes wbot1 carrier1 box2 box1 corridor1 reception)
      [5.000]
19.005: (deliver_patient hbot1 patient1 operation_unit surgical_block) [1.000]
20.006: (move_bot_alone hbot1 surgical_block corridor1) [3.000]
22.008: (move_worker_bot_with_2_boxes wbot1 carrier1 box2 box1 reception
      emergency_room) [5.000]
23.007: (move_bot_alone hbot1 corridor1 reception) [3.000]
26.008: (move_bot_alone hbot1 reception entrance) [3.000]
27.009: (drop_down_2nd_box wbot1 carrier1 box2) [2.000]
29.009: (move_bot_alone hbot1 entrance reception) [3.000]
29.010: (deliver_supply wbot1 box2 scalpel emergency_room trauma_unit) [1.000]
30.011: (drop_down_1st_box wbot1 carrier1 box1) [2.000]
32.010: (move_bot_alone hbot1 reception corridor1) [3.000]
32.012: (deliver_supply wbot1 box1 defibrillator emergency_room resuscitation_unit)
      [1.000]
33.013: (pick_up_1st_box wbot1 carrier1 box2 emergency_room) [2.000]
35.011: (move_bot_alone hbot1 corridor1 surgical_block) [3.000]
35.014: (pick_up_2nd_box wbot1 carrier1 box1 emergency_room) [2.000]
37.015: (move_worker_bot_with_2_boxes wbot1 carrier1 box2 box1 emergency_room
      reception) [5.000]
38.012: (pick_up_patient hbot1 patient1 surgical_block) [2.000]
40.013: (move_helper_bot_patient hbot1 surgical_block corridor1 patient1) [5.000]
42.016: (move_worker_bot_with_2_boxes wbot1 carrier1 box2 box1 reception corridor1)
      [5.000]
45.014: (move_helper_bot_patient hbot1 corridor1 reception patient1) [5.000]
47.017: (move_worker_bot_with_2_boxes wbot1 carrier1 box2 box1 corridor1
      central_warehouse) [5.000]
50.015: (drop_down_patient hbot1 patient1) [2.000]
52.016: (move_bot_alone hbot1 reception entrance) [3.000]
52.018: (drop_down_2nd_box wbot1 carrier1 box2) [2.000]
54.019: (drop_down_1st_box wbot1 carrier1 box1) [2.000]
```

Listing 11: `p4_0` OPTIC output

```
; Plan found with metric 123.040
; States evaluated so far: 22974
; States pruned based on pre-heuristic cost lower bound: 0
; Time 114.96
0.000: (fill_box wbot1 box1 surgical_mask central_warehouse) [2.000]
0.000: (pick_up_patient hbot1 patient2 entrance) [2.000]
0.000: (move_bot_alone hbot2 entrance reception) [3.000]
0.001: (fill_box wbot2 box1 stethoscope central_warehouse) [2.000]
2.001: (fill_box wbot1 box2 scalpel central_warehouse) [2.000]
2.001: (move_helper_bot_patient hbot1 entrance reception patient2) [5.000]
2.002: (move_bot_alone wbot2 central_warehouse corridor1) [3.000]
3.001: (move_bot_alone hbot2 reception corridor1) [3.000]
4.002: (pick_up_1st_box wbot1 carrier1 box1 central_warehouse) [2.000]
5.003: (move_bot_alone wbot2 corridor1 surgical_block) [3.000]
6.002: (move_bot_alone hbot2 corridor1 diagnostic_lab) [3.000]
6.003: (fill_box wbot1 box3 defibrillator central_warehouse) [2.000]
7.002: (drop_down_patient hbot1 patient2) [2.000]
8.004: (pick_up_2nd_box wbot1 carrier1 box4 central_warehouse) [2.000]
8.004: (move_bot_alone wbot2 surgical_block corridor1) [3.000]
9.003: (pick_up_patient hbot2 patient3 diagnostic_lab) [2.000]
9.003: (pick_up_patient hbot1 patient2 reception) [2.000]
10.005: (drop_down_2nd_box wbot1 carrier1 box1) [2.000]
11.004: (drop_down_patient hbot2 patient3) [2.000]
11.004: (drop_down_patient hbot1 patient2) [2.000]
11.005: (move_bot_alone wbot2 corridor1 surgical_block) [3.000]
12.006: (pick_up_2nd_box wbot1 carrier1 box3 central_warehouse) [2.000]
13.005: (pick_up_patient hbot2 patient3 diagnostic_lab) [2.000]
13.005: (pick_up_patient hbot1 patient2 reception) [2.000]
14.006: (move_bot_alone wbot2 surgical_block corridor1) [3.000]
14.007: (pick_up_3rd_box wbot1 carrier1 box2 central_warehouse) [2.000]
15.006: (move_helper_bot_patient hbot1 reception corridor1 patient2) [5.000]
15.006: (move_helper_bot_patient hbot2 diagnostic_lab neurology_ward patient3)
      [5.000]
16.008: (drop_down_3rd_box wbot1 carrier1 box4) [2.000]
17.007: (move_bot_alone wbot2 corridor1 central_warehouse) [3.000]
18.009: (pick_up_3rd_box wbot1 carrier1 box1 central_warehouse) [2.000]
```

```
20.007: (move_helper_bot_patient hbot1 corridor1 surgical_block patient2) [5.000]
20.007: (drop_down_patient hbot2 patient3) [2.000]
20.008: (fill_box wbot2 box4 scalpel central_warehouse) [2.000]
20.010: (move_worker_bot_with_3_boxes wbot1 carrier1 box3 box2 box1
      central_warehouse corridor1) [5.000]
22.008: (deliver_patient hbot2 patient3 neurology_unit neurology_ward) [1.000]
22.009: (move_bot_alone wbot2 central_warehouse corridor1) [3.000]
23.009: (move_bot_alone hbot2 neurology_ward diagnostic_lab) [3.000]
25.008: (move_helper_bot_patient hbot1 surgical_block cardiology_ward patient2)
      [5.000]
25.010: (move_bot_alone wbot2 corridor1 reception) [3.000]
25.011: (move_worker_bot_with_3_boxes wbot1 carrier1 box3 box2 box1 corridor1
      reception) [5.000]
26.010: (move_bot_alone hbot2 diagnostic_lab corridor1) [3.000]
28.011: (move_bot_alone wbot2 reception emergency_room) [3.000]
29.011: (move_bot_alone hbot2 corridor1 reception) [3.000]
30.009: (drop_down_patient hbot1 patient2) [2.000]
30.012: (drop_down_3rd_box wbot1 carrier1 box2) [2.000]
31.012: (move_bot_alone wbot2 emergency_room reception) [3.000]
32.010: (deliver_patient hbot1 patient2 cardiology_unit cardiology_ward) [1.000]
32.012: (move_bot_alone hbot2 reception entrance) [3.000]
32.013: (move_worker_bot_with_2_boxes wbot1 carrier1 box3 box1 reception
      emergency_room) [5.000]
33.011: (move_bot_alone hbot1 cardiology_ward surgical_block) [3.000]
34.013: (pick_up_1st_box wbot2 carrier2 box2 reception) [2.000]
35.013: (pick_up_patient hbot2 patient1 entrance) [2.000]
36.014: (move_worker_bot_with_1_box wbot2 carrier2 box2 reception emergency_room)
      [5.000]
37.014: (move_helper_bot_patient hbot2 entrance reception patient1) [5.000]
37.014: (move_worker_bot_with_2_boxes wbot1 carrier1 box3 box1 emergency_room
      reception) [5.000]
41.015: (drop_down_1st_box wbot2 carrier2 box2) [2.000]
42.015: (move_helper_bot_patient hbot2 reception corridor1 patient1) [5.000]
42.015: (drop_down_2nd_box wbot1 carrier1 box1) [2.000]
43.016: (move_bot_alone wbot2 emergency_room reception) [3.000]
45.017: (pick_up_2nd_box wbot1 carrier1 box1 reception) [2.000]
46.017: (deliver_supply wbot2 box1 surgical_mask reception triage_unit) [1.000]
47.016: (move_helper_bot_patient hbot2 corridor1 surgical_block patient1) [5.000]
47.018: (move_bot_alone wbot2 reception emergency_room) [3.000]
47.018: (move_worker_bot_with_2_boxes wbot1 carrier1 box3 box1 reception corridor1)
      [5.000]
50.019: (deliver_supply wbot2 box2 scalpel emergency_room trauma_unit) [1.000]
51.020: (move_bot_alone wbot2 emergency_room reception) [3.000]
52.017: (drop_down_patient hbot2 patient1) [2.000]
52.019: (move_worker_bot_with_2_boxes wbot1 carrier1 box3 box1 corridor1 reception)
      [5.000]
54.017: (deliver_patient hbot1 patient1 operation_unit surgical_block) [1.000]
54.021: (move_bot_alone wbot2 reception corridor1) [3.000]
57.020: (drop_down_2nd_box wbot1 carrier1 box1) [2.000]
57.022: (move_bot_alone wbot2 corridor1 surgical_block) [3.000]
59.021: (move_worker_bot_with_1_box wbot1 carrier1 box3 reception emergency_room)
      [5.000]
60.023: (move_bot_alone wbot2 surgical_block corridor1) [3.000]
63.024: (move_bot_alone wbot2 corridor1 reception) [3.000]
64.022: (drop_down_1st_box wbot1 carrier1 box3) [2.000]
66.023: (deliver_supply wbot1 box3 defibrillator emergency_room resuscitation_unit)
      [1.000]
66.025: (pick_up_1st_box wbot2 carrier2 box1 reception) [2.000]
67.024: (move_bot_alone wbot1 emergency_room reception) [3.000]
68.026: (move_worker_bot_with_1_box wbot2 carrier2 box1 reception corridor1) [5.000]
70.025: (move_bot_alone wbot1 reception corridor1) [3.000]
73.026: (move_bot_alone wbot1 corridor1 diagnostic_lab) [3.000]
73.027: (move_worker_bot_with_1_box wbot2 carrier2 box1 corridor1 central_warehouse)
      [5.000]
76.027: (move_bot_alone wbot1 diagnostic_lab neurology_ward) [3.000]
78.028: (drop_down_1st_box wbot2 carrier2 box1) [2.000]
79.028: (move_bot_alone wbot1 neurology_ward diagnostic_lab) [3.000]
80.029: (fill_box wbot2 box1 blood_bag central_warehouse) [2.000]
82.029: (move_bot_alone wbot1 diagnostic_lab neurology_ward) [3.000]
82.030: (pick_up_1st_box wbot2 carrier2 box1 central_warehouse) [2.000]
84.031: (move_worker_bot_with_1_box wbot2 carrier2 box1 central_warehouse corridor1)
      [5.000]
85.030: (move_bot_alone wbot1 neurology_ward diagnostic_lab) [3.000]
88.031: (move_bot_alone wbot1 diagnostic_lab corridor1) [3.000]
89.032: (drop_down_1st_box wbot2 carrier2 box1) [2.000]
91.033: (pick_up_1st_box wbot2 carrier2 box1 corridor1) [2.000]
91.034: (pick_up_1st_box wbot1 carrier1 box1 corridor1) [2.000]
93.034: (drop_down_1st_box wbot2 carrier2 box1) [2.000]
93.035: (move_worker_bot_with_1_box wbot1 carrier1 box1 corridor1 surgical_block)
      [5.000]
95.035: (move_bot_alone wbot2 corridor1 surgical_block) [3.000]
98.036: (move_worker_bot_with_1_box wbot1 carrier1 box1 surgical_block
      cardiology_ward) [5.000]
98.036: (move_bot_alone wbot2 surgical_block cardiology_ward) [3.000]
101.037: (deliver_supply wbot2 box1 stethoscope cardiology_ward cardiology_unit)
      [1.000]
102.038: (move_bot_alone wbot2 cardiology_ward surgical_block) [3.000]
103.037: (move_worker_bot_with_1_box wbot1 carrier1 box1 cardiology_ward
      surgical_block) [5.000]
105.039: (move_bot_alone wbot2 surgical_block corridor1) [3.000]
108.038: (move_worker_bot_with_1_box wbot1 carrier1 box1 surgical_block corridor1)
      [5.000]
108.040: (move_bot_alone wbot2 corridor1 diagnostic_lab) [3.000]
111.041: (move_bot_alone wbot2 diagnostic_lab neurology_ward) [3.000]
113.039: (move_worker_bot_with_1_box wbot1 carrier1 box1 corridor1 diagnostic_lab)
      [5.000]
114.042: (move_bot_alone wbot2 neurology_ward diagnostic_lab) [3.000]
117.043: (move_bot_alone wbot2 diagnostic_lab neurology_ward) [3.000]
118.040: (move_worker_bot_with_1_box wbot1 carrier1 box1 diagnostic_lab
      neurology_ward) [5.000]
120.044: (deliver_supply wbot2 box1 blood_bag neurology_ward neurology_unit) [1.000]
```

Listing 12: `p4_1` OPTIC output

```
[INFO] [plansys2_node-1]: process started with pid [5259]
[INFO] [deliver_patient_node-2]: process started with pid [5261]
[INFO] [deliver_supply_node-3]: process started with pid [5263]
[INFO] [drop_down_1st_box_node-4]: process started with pid [5265]
[INFO] [drop_down_2nd_box_node-5]: process started with pid [5267]
[INFO] [drop_down_3rd_box_node-6]: process started with pid [5269]
[INFO] [drop_down_patient_node-7]: process started with pid [5271]
[INFO] [fill_box_node-8]: process started with pid [5273]
[INFO] [move_worker_bot_alone_node-9]: process started with pid [5275]
[INFO] [move_helper_bot_alone_node-10]: process started with pid [5277]
[INFO] [move_helper_bot_patient_node-11]: process started with pid [5279]
[INFO] [move_worker_bot_with_1_box_node-12]: process started with pid [5281]
[INFO] [move_worker_bot_with_2_boxes_node-13]: process started with pid [5283]
[INFO] [move_worker_bot_with_3_boxes_node-14]: process started with pid [5307]
[INFO] [pick_up_1st_box_node-15]: process started with pid [5369]
[INFO] [pick_up_2nd_box_node-16]: process started with pid [5373]
[INFO] [pick_up_3rd_box_node-17]: process started with pid [5415]
[INFO] [pick_up_patient_node-18]: process started with pid [5418]
...
... Other instructions ....
...
pick_up_1st_box ... [100%]
pick_up_patient ... [100%]
pick_up_2nd_box ... [100%]
move_helper_bot_patient ... [100%]
[plansys2_node-1] [WARN] [1753803269.858334588] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
drop_down_2nd_box ... [100%]
move_helper_bot_patient ... [100%]
[plansys2_node-1] [WARN] [1753803282.998356757] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
fill_box ... [100%]
drop_down_patient ... [100%]
pick_up_2nd_box ... [100%]
pick_up_patient ... [100%]
drop_down_2nd_box ... [100%]
move_helper_bot_patient ... [100%]
[plansys2_node-1] [WARN] [1753803322.260170897] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
move_worker_bot_with_1_box ... [100%]
drop_down_patient ... [100%]
[plansys2_node-1] [WARN] [1753803335.398108925] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
move_worker_bot_with_1_box ... [100%]
deliver_patient ... [100%]
[plansys2_node-1] [WARN] [1753803348.457939229] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
move_worker_bot_with_1_box ... [100%]
move_helper_bot_alone ... [100%]
[plansys2_node-1] [WARN] [1753803361.572864295] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
[plansys2_node-1] [WARN] [1753803361.577162569] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
[plansys2_node-1] [WARN] [1753803361.579303521] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
drop_down_1st_box ... [100%]
move_helper_bot_alone ... [100%]
deliver_supply ... [100%]
move_helper_bot_alone ... [100%]
[plansys2_node-1] [WARN] [1753803387.768476187] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
pick_up_1st_box ... [100%]
[plansys2_node-1] [WARN] [1753803400.858838755] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
move_worker_bot_with_1_box ... [100%]
[plansys2_node-1] [WARN] [1753803413.971182384] [LifecyclePublisher]: Trying to
      publish message on the topic '/actions_hub', but the publisher is not
      activated
move_worker_bot_with_1_box ... [100%]
move_worker_bot_with_1_box ... [100%]
drop_down_1st_box ... [100%]
fill_box ... [100%]
pick_up_1st_box ... [100%]
move_worker_bot_with_1_box ... [100%]
move_worker_bot_with_1_box ... [100%]
move_worker_bot_with_1_box ... [100%]
drop_down_1st_box ... [100%]
deliver_supply ... [100%]
pick_up_1st_box ... [100%]
move_worker_bot_with_1_box ... [100%]
move_worker_bot_with_1_box ... [100%]
drop_down_1st_box ... [100%]
move_worker_bot_alone ... [100%]
pick_up_1st_box ... [100%]
move_worker_bot_with_1_box ... [100%]
pick_up_2nd_box ... [100%]
move_worker_bot_with_2_boxes ... [100%]
drop_down_2nd_box ... [100%]
drop_down_1st_box ... [100%]
[plansys2_node-1] [INFO] [1753803689.167823877] [executor]: Plan Succeeded
```

Listing 13: `p5_0` Terminal 1

```
root@luca-XPS-15-7590:/workspace/p5# ros2 launch p5 p5_launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log
      /2025-07-29-16-54-56-649679-luca-XPS-15-7590-5258
[INFO] [launch]: Default logging verbosity is set to INFO
```

```
root@luca-XPS-15-7590:/workspace/p5# ros2 run plansys2_terminal plansys2_terminal
[INFO] [1753804503.835544547] [terminal]: No problem file specified.
ROS2 Planning System console. Type "quit" to finish
> source /workspace/p5/launch/commands
> get plan
```

```
plan:
0: (pick_up_1st_box w_bot1 carrier1 box1 central_warehouse) [2]
0: (pick_up_patient h_bot1 patient1 entrance) [2]
2.001: (move_helper_bot_patient h_bot1 entrance reception patient1) [5]
2.001: (pick_up_2nd_box w_bot1 carrier1 box2 central_warehouse) [2]
4.002: (drop_down_2nd_box w_bot1 carrier1 box1) [2]
6.003: (fill_box w_bot1 box1 defibrillator central_warehouse) [2]
7.002: (move_helper_bot_patient h_bot1 reception corridor1 patient1) [5]
8.004: (pick_up_2nd_box w_bot1 carrier1 box1 central_warehouse) [2]
10.005: (drop_down_2nd_box w_bot1 carrier1 box2) [2]
12.003: (drop_down_patient h_bot1 patient1) [2]
12.006: (move_worker_bot_with_1_box w_bot1 carrier1 box1 central_warehouse corridor1
       ) [5]
14.004: (pick_up_patient h_bot1 patient1 corridor1) [2]
16.005: (move_helper_bot_patient h_bot1 corridor1 surgical_block patient1) [5]
17.007: (move_worker_bot_with_1_box w_bot1 carrier1 box1 corridor1 reception) [5]
21.006: (drop_down_patient h_bot1 patient1) [2]
22.008: (move_worker_bot_with_1_box w_bot1 carrier1 box1 reception emergency_room)
       [5]
23.007: (deliver_patient h_bot1 patient1 operation_unit surgical_block) [1]
24.008: (move_helper_bot_alone h_bot1 surgical_block corridor1) [3]
27.009: (move_helper_bot_alone h_bot1 corridor1 reception) [3]
27.009: (drop_down_1st_box w_bot1 carrier1 box1) [2]
29.01: (deliver_supply w_bot1 box1 defibrillator emergency_room resuscitation_unit)
       [1]
30.01: (move_helper_bot_alone h_bot1 reception entrance) [3]
30.011: (pick_up_1st_box w_bot1 carrier1 box1 emergency_room) [2]
32.012: (move_worker_bot_with_1_box w_bot1 carrier1 box1 emergency_room reception)
       [5]
37.013: (move_worker_bot_with_1_box w_bot1 carrier1 box1 reception corridor1) [5]
42.014: (move_worker_bot_with_1_box w_bot1 carrier1 box1 corridor1 central_warehouse
       ) [5]
47.015: (drop_down_1st_box w_bot1 carrier1 box1) [2]
49.016: (fill_box w_bot1 box1 scalpel central_warehouse) [2]
51.017: (pick_up_1st_box w_bot1 carrier1 box1 central_warehouse) [2]
53.018: (move_worker_bot_with_1_box w_bot1 carrier1 box1 central_warehouse corridor1
       ) [5]
58.019: (move_worker_bot_with_1_box w_bot1 carrier1 box1 corridor1 reception) [5]
63.02: (move_worker_bot_with_1_box w_bot1 carrier1 box1 reception emergency_room)
       [5]
68.021: (drop_down_1st_box w_bot1 carrier1 box1) [2]
70.022: (deliver_supply w_bot1 box1 scalpel emergency_room trauma_unit) [1]
71.023: (pick_up_1st_box w_bot1 carrier1 box1 emergency_room) [2]
73.024: (move_worker_bot_with_1_box w_bot1 carrier1 box1 emergency_room reception)
       [5]
78.025: (move_worker_bot_with_1_box w_bot1 carrier1 box1 reception corridor1) [5]
83.026: (drop_down_1st_box w_bot1 carrier1 box1) [2]
85.027: (move_worker_bot_alone w_bot1 corridor1 central_warehouse) [3]
88.028: (pick_up_1st_box w_bot1 carrier1 box2 central_warehouse) [2]
90.029: (move_worker_bot_with_1_box w_bot1 carrier1 box2 central_warehouse corridor1
       ) [5]
95.03: (pick_up_2nd_box w_bot1 carrier1 box1 corridor1) [2]
97.031: (move_worker_bot_with_2_boxes w_bot1 carrier1 box2 box1 corridor1
       central_warehouse) [5]
102.032: (drop_down_2nd_box w_bot1 carrier1 box1) [2]
104.033: (drop_down_1st_box w_bot1 carrier1 box2) [2]
> run
[(pick_up_1st_box w_bot1 carrier1 box1 central_warehouse) 0%]
[(pick_up_1st_box w_bot1 carrier1 box1 central_warehouse) 2%]
...
... The Plan executes through fake actions ....
...
Successful finished
```

Listing 14: `p5_0` Terminal 2