On Bringing HTN Domains Closer to Reality
The Case of Satellite and Rover Domains

Ebaa Alnazer, Ilche Georgievski, Marco Aiello
Service Computing Department, Institute of Architecture of Application Systems, University of Stuttgart
firstname.lastname@iaas.uni-stuttgart.de

Abstract

Actual real-world domains are characterised by offering many choices of how agents populating these domains can achieve certain objectives. A widely used technique for planning in real-world domains is Hierarchical Task Network (HTN) planning. This technique supports rich domain knowledge using tasks, decomposition methods and high expressive power, allowing agents to search for plans quickly. In practice, however, benchmark HTN domains have a tendency to oversimplify some real-world aspects to enable planners to find plans and evaluate their performance. For example, most benchmark HTN domains do not model a wide range of alternative choices typical of real-world domains; the modelling only restricts the choice of a particular method when decomposing a task in a particular state. In this paper, we provide a set of realistic domain aspects, analyse the benchmark HTN domains considering some of them, and suggest extensions to specific benchmark HTN domains to address alternative choices, thus reflecting upon some of the aspects. We believe that the suggested extensions and similar ones do not only bring the benchmarks closer to reality, but also contribute an important ingredient for future HTN unifying frameworks and for improving the quality of HTN planners.

1 Introduction

The idea of enriching planning domains with knowledge on how to accomplish tasks dates back to 1975 when Sacerdoti proposed the hierarchical structure of procedural nets (Sacerdoti 1975). With major contributions made by SIPE-2 (Wilkins 1988), O-Plan (Currie and Tate 1991), UMCP (Erol, Hendler, and Nau 1994) and SHOP (Nau et al. 1999), this idea eventually resulted in what we know as Hierarchical Task Network (HTN) planning (Georgievski and Aiello 2015; Bercher, Alford, and Höller 2019). HTN planning is today a proven technique for solving planning problems quickly. A huge part of the speed of computation comes from the hierarchical structure of HTNs that encode the domain knowledge and that are used to search for plans.

Planning approaches are typically tested, evaluated and compared among each other using benchmark domains developed for and used in the International Planning Competitions (IPCs). In the context of HTN planning, a collection of benchmark HTN domains have been collected only recently, during IPC 2020. Potential contributors were encouraged to submit domains that go beyond the capabilities of classical planners and can exploit the high expressive power of HTN planning (Behnke et al. 2019). In the end, HTN domains were individually designed or translated from benchmark domains for non-hierarchical planning. This collection represents a first step towards a unifying evaluation framework for HTN planners (Behnke, Höller, and Bercher 2021).

While the main advantage of HTN planning is its support for rich domain knowledge and high expressive power, there seems to be a risk of oversimplifying the benchmark HTN domains. For example, HTN planning offers the possibility to include several, alternative ways of achieving a task by using methods. However, most benchmark HTN domains do not model alternative ways of performing tasks. Such a simplification might be useful or needed so that the benchmark HTN domains can be handled by existing planning approaches within the scope of IPC. However, this might not be that useful when we need realistic domains to test and evaluate planning approaches for actual application needs. In this context, we often have criteria on which alternative to choose, which eventually affects the quality of plans. Thus, the lack of alternatives affects not only the possibility to choose but also the plan quality. Moreover, since some of the benchmark HTN domains are translated from classical planning domains, which were created years and even decades ago, they no longer keep up with the current trends and advancements of technology in the corresponding real-world domains, making them obsolete and of little practical use.

Our work stems from the need to move some of the HTN domains a step closer to reality, not necessarily for the sole purpose of IPC. We describe a non-exhaustive set of aspects that can make HTN domains more realistic and that can be considered when designing new domains or improving existing ones. We report about, present and model new aspects in two of the benchmark domains, namely Satellite and Rover.

The contributions of this paper are: (1) a set of realistic domain aspects; (2) the analysis of current benchmark HTN domains in terms of available alternatives; and (3) extensions of the Satellite and Rover domains to increase how realistic they are. Apart from this, our work has value as

Copyright © 2022. All rights reserved.

an example to bring some of the benchmark HTN domains closer to real-world settings. Our ultimate objective is to use such domains to not only exploit the full potential of HTN planning but also to foster the development of new concepts and mechanisms for HTN planning.

The remainder of the paper is organised as follows. Section 2 provides a motivating example of a real-world domain followed by Section 3 that presents the realistic domain aspects. Section 4 provides an overview of the current benchmark HTN domains. Section 5 introduces the extensions of the Satellite and Rover domains. Sections 6 and 7 close the paper with a discussion and conclusions, respectively.

2 A Motivating Example

We start by providing an example of a common planning activity in the travelling domain. A person wants to plan a trip, which involves tasks like booking an accommodation, booking tickets for the transportation means, etc. Each step in this planning process involves alternative choices that the person can make in a specific *state of the environment*. For example, the state of the environment at the beginning is that accommodation is not booked and tickets are not bought yet. In this state and as a first step, the person has to decide which type of accommodation to book, e.g., whether go for a hotel or Airbnb. Say the person chooses a hotel accommodation. Then, the person has to decide which type of room to book.

After booking the hotel room, the person is in a new state where the person has a hotel but not transportation tickets. So, the person has to decide between travelling by plane or train. Say that the person chose to go by train. To book tickets for the train, the person can make other choices. For example, the person should decide on the type of train, then the class, followed by a selection of the seat position and seat number. After booking the train ticket, there might be other decisions to be made, such as booking a transport from the train station to the hotel, for which similar reasoning can be applied when faced with alternatives and choices.

This simple example contains two important features. The first one is the *hierarchy of tasks and decisions*, and the second one are the *alternatives* available at the same time at each hierarchical level on how to perform a specific activity. In this domain, the person can be offered unconstrained alternatives and choices to choose from. For example, the choice of booking a plane or a train. Other alternatives can be constrained with special conditions and depend on the state of the environment. For example, the person can take either a train or a taxi to get to the hotel. In the first case, the person has to have train tickets, while in the second case, s/he should have money. If the person has only train tickets, but no money, the only option is to take the train. However, if the person has both, s/he can freely choose an option.

3 Towards Realistic HTN Domains

To bring a domain closer to reality, one can consider the common and interrelated aspects that real-world domains are characterised by. We refer to these as *realistic domain aspects*. We systematise a list of realistic domain aspects by collecting real-world concepts, intuitions and dimensions accepted for classical benchmark domains and AI planning in general (Hoffmann et al. 2006; Georgievski and Aiello 2016). Other aspects are specific to HTN planning and are insights drawn from our own analysis of hierarchical structures.

Aspect 1: Realism. HTN domains should describe an actual or possible application of planning technology, rather than modelling oversimplified or toy applications.

Aspect 2: Structural diversity. HTN domains should be structurally diverse. That means, each planning domain should model the specificities of the real-world domain it represents instead of just re-stating the same tasks of other similar domains. For example, many domains, such as *Logistics*, *Rover*, and *Transport*, model the tasks of moving between locations. For these domains, instead of just restating the tasks related to the movement, the domains should also consider modelling knowledge specific to the domain. For example, our motivating example has similar tasks that can exist in other domains (e.g., the moving task). However, it has many specific tasks that do not exist in other domains, such as booking attractions and tours at the destination.

Aspect 3: Alternatives. HTN domains should model a wide range of alternatives to achieve tasks. This is a property that exists when planning in many real-world domains. For example, in the motivating example, at each planning step, multiple alternatives are available to the person planning the trip. In HTN planning, methods are used to model alternatives for achieving tasks. We propose a categorisation of methods according to their possibility of forming an alternative to other method(s), namely *strict alternative*, *weak alternative*, and *no alternative*. The methods are strict alternatives to each other if they all have the same components that do not contradict in their preconditions. However, if two or more methods have preconditions that do not have the same components, but are not mutually exclusive, they are weak alternatives to each other. This is because the domain cannot solely determine whether they will be applicable all the same time during planning. This is only determined during the planning when binding the variables to specific objects from the problem instance. A method forms no alternative to (an)other method(s), if it is the only method that can decompose a particular task, or if its preconditions contradict with the preconditions of other method(s) of this task. Considering the motivating example, the alternatives given for getting to the hotel are weak alternatives. That is, their applicability at the same time is only determined during planning based on whether the person has tickets and money in the same state. The case of having unconstrained alternatives of booking a plane or train illustrates strict alternatives.

Aspect 4: Operator costs. HTN domains should encode operator/action costs to represent the fact that performing actions in real world domains entails costs to be bore, such as fuel, money, time, and effort. For instance, Figure 1 shows an example of possible alternatives for going from the train

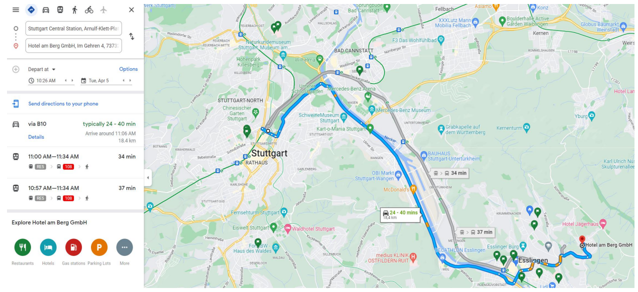

Figure 1: A screenshot taken from Google maps of alternative transportation means with different traveling times between Stuttgart main station and a hotel.

station to the hotel, where each alternative has a different cost (time in this example). Driving the car, in this example, will take 24-40 minutes, while taking a regional train then a bus takes 34 minutes or 37 minutes. If the person cares about reaching the hotel as fast as possible, s/he needs to take the travelling time into account when making decisions.

Aspect 5: Numeric characteristics. Several aspects in real-world domains, e.g., temperature, are expressed numerically over a potentially infinite range of values. Thus, in order for the HTN domains to reflect the real-world settings more precisely, they should model numeric variables.

Aspect 6: Temporal characteristics. Performing actions in real-world domains requires time; and planning in these domains involves temporal reasoning about these actions. In particular, there can exist temporal constraints that restrict the ordering between the execution of different actions. For instance, in the motivating example, the action of selecting a seat in the train cannot start before the end of the selection of a specific train. Moreover, some actions in real-world domains have to be executed concurrently (Cushing et al. 2007). For instance, the travelling domain might require concurrent actions for booking a train and a hotel.

Aspect 7: Evolving technological awareness. HTN domains should consider, when necessary, the recent technologies and trends that exist or could possibly exist in real-world domains. In the motivating example, the domain model should take into account the possibility of booking an Airbnb, if available, as an alternative to booking a hotel room for accommodation.

4 Benchmark HTN Domains

Benchmark domains are used in the planning field to test planning approaches and compare them to one another. They are an important instrument for the field as they can help driving the development of future approaches (Hoffmann et al. 2006). In this context, a set of benchmark domains for HTN planning is provided in the International Planning Competition of 2020 (IPC-2020). These domains represent a step forward to the establishment of a common and unified framework for comparing HTN planners.

The HTN benchmark domains are organised in two categories based on the type of ordering in task networks, namely partial order and total order. A domain in the partial-order category has at least one method whose subtasks are not totally ordered or the initial task network is not a sequence. A domain in the total-order category has methods with totally ordered subtasks in their task networks and the initial task network forms a sequence. The partial-order category contains 9 domains, while the total-order category includes 24 domains. In addition to these domains, there are also other domains that are submitted, but not included in the competition. One of the reasons for the exclusion is that these domains require expressiveness not supported by the HTN planners participating in the competition.

To gain more insights about the benchmark domains, we can analyse them in terms of their structural diversity. However, we do not want to compare the specificities of each domain with the specificities of another domain since our goal is not to compare the domains among each other, but to gain insights about the different aspects each domain offers. So, we look at the structural diversity using the following *structural metrics*: (1) total number of methods **#m**; (2) number of methods that are strict alternatives **#ms**; (3) number of methods that are weak alternatives **#mw**; (4) number of methods that do not form an alternative **#mn**; (5) number of tasks **#t**; (6) number of direct recursive tasks **#rt**, where the direct recursive task is the task that is contained in the task network of one of the methods that can decompose it; and (7) number of methods that have x subtasks **#m-tn**.[1]

Table 1 shows the results of applying the structural metrics to 20 representative benchmark HTN domains. The first four metrics quantify all types of alternative methods. Most notably, we see that most domains do not offer strict alternative methods. Moreover, comparing the number of "no alternative" methods with the number of weak alternatives, we see that in most domains the former is bigger. Also, a few domains do not offer alternatives at all. This indicates that the range of planning choices that most domains offer is rather limiting and might not entirely reflect the range of alternatives that exist in their real-world counterparts. Additionally, most domains contain direct recursive tasks and the only domains that they do not contain recursive tasks at all are Barman-BDI, Childsnack, and Woodworking as also reported in (Behnke, Höller, and Bercher 2021). Finally, we notice that the majority of domains have methods with task networks of one or two tasks, meaning the methods decompose their corresponding tasks into a single or two subtasks.

[1]Expressiveness can also be an indication of the structural diversity. However, since our extensions do not focus on realistic domain aspects that require more expressiveness, such as **Aspect 4**, **Aspect 5** and **Aspect 6**, related metrics are not considered. Incidentally, in IPC-2020, the expressiveness constructs were limited to (negated) literals, conjunctions, and `forall` universal quantifier, so no valuable insights can be gathered about those three aspects.

Table 1: Insights into benchmark HTN domains.

Domain	#m	#ms	#mw	#mn	#t	#rt	#m-tn 1	#m-tn 2	#m-tn 3	#m-tn 4	#m-tn 5	#m-tn 6	#m-tn >6
Total-Order Category													
AssemblyHierarchical	16	10	0	6	4	1	13	3	0	0	0	0	0
Barman-BDI	13	0	4	9	10	0	3	0	5	4	0	1	0
Blocksworld-GTOHP	5	0	0	5	4	0	0	3	1	1	0	0	0
Blocksworld-HPDDL	11	0	8	3	5	1	0	5	1	5	0	0	0
Childsnack	2	0	0	2	1	0	0	0	0	0	2	0	0
Depots	10	0	5	5	6	1	2	1	2	4	0	0	0
Elevator-L	22	0	0	22	12	1	12	7	2	0	1	0	0
Entertainment	24	20	3	1	11	3	16	7	1	0	0	1	1
Factories-s	7	0	2	5	5	3	1	3	2	1	0	0	0
Hiking	11	0	2	9	8	1	3	7	0	0	0	0	1
Logistics-L	37	0	6	31	14	5	15	19	4	0	0	0	0
Minecraft-P	17	0	5	12	8	2	7	9	0	0	0	1	0
Minecraft-R	12	0	4	8	7	3	5	6	0	0	0	1	0
Multiarm-Blocksworld	11	0	5	6	5	1	4	5	0	2	1	0	0
Rover-G	13	0	0	13	10	1	4	4	1	4	0	0	0
Satellite-G	8	0	0	8	6	0	2	4	1	0	0	0	0
Snake	3	0	0	3	2	2	0	2	1	0	0	0	0
Towers	7	0	2	5	5	1	4	3	0	0	0	0	0
Transport	4	0	2	2	4	1	3	1	0	0	0	0	0
Woodworking	19	15	4	0	6	0	14	2	3	0	0	0	0
Partial-Order Category													
Domain	**#m**	**#ms**	**#mw**	**#mn**	**#t**	**#rt**	**#m-tn** 1	2	3	4	5	6	>6
Barman-BDI	13	0	4	9	10	0	3	0	5	4	0	1	0
Monroe-FO	62	32	2	28	40	1	20	20	12	3	4	5	0
Monroe-PO	62	32	2	28	41	1	19	20	12	3	4	5	0
PCP	12	12	0	0	2	2	1	2	4	5	0	0	0
Rover	11	0	0	11	9	0	2	4	1	4	0	0	0
Satellite	8	0	4	4	3	0	2	4	2	0	0	0	0
Transport	4	0	2	2	4	1	3	1	0	0	0	0	0
UM-Translog	51	16	28	7	21	18	8	15	5	5	4	0	0
Woodworking	19	15	4	0	6	0	14	2	3	0	0	0	0

5 Extensions for Realism, Structural Diversity and Alternatives

Our approach to bringing benchmark domains closer to real-world settings focuses on addressing **Aspect 1**, **Aspect 2**, **Aspect 3**, and **Aspect 7**. In particular, we address the lack of alternatives problem by adding more ways to achieve tasks, i.e., more methods. The methods that can achieve the same task should have preconditions that allow them to be all applicable to decompose the task in a particular state of the world. Adding these methods contributes to the fulfillment of **Aspect 3**. Moreover, these alternatives should be modeled in domains that are built for a real planning application, and should contribute to the realism of these domains (**Aspect 1**). They should also contribute to the structural diversity of the domain by modeling specificities in that domain (**Aspect 2**). Finally, they should consider the up-to-date trends and technologies in the domain they model (**Aspect 7**).

To demonstrate the possibility of extension, we select two representative benchmark domains for extension. Both domains are originally built from actual applications of planning technology. The first domain is *Satellite-GTOHP* from the total-order category (Pellier and Fiorino 2021). From here on, we refer to this domain as *Satellite*. The second domain is *Rover* from the partial-order category (Höller et al. 2018). The domain is a translation of the domain the deployment of which is presented in (Nau et al. 2003).

For each of the domains, we provide the following: a description of the **application** domain; a **motivation** behind our extension in terms of the realistic domain aspects; a description of the domain **extension**; a description of how this extension addresses one or more of the **realistic domain aspects**; and excerpts of the **encoding** of the extension.[2] The encoding is specified in the Hierarchical Domain Definition Language (HDDL) (Höller et al. 2020).

[2]The full domain models are available on Github: https://github.com/planningGeek/Rover-Satellite-domains-extended.

5.1 Satellite Domain

Application. This domain models a NASA space application, where satellites have missions of making observations by taking images of various spatial phenomena in varying signal frequencies, e.g., infrared, spectrograph, x-ray, and thermograph. Each satellite is equipped with observation instruments, each of which supports specific modes with defined calibration targets (directions). Accomplishing missions, i.e., making observations, involves preparing a satellite and then taking an image. Preparing the satellite involves two tasks. The first task is achieved by routing energy to an instrument while ensuring only one instrument is powered on a particular satellite and then properly calibrating the instrument. The second task is achieved by turning the satellite to the direction of the phenomenon to be captured. Taking an image is an action that can be executed directly.

Motivation. The Satellite domain does not model the most important task that constitutes the main reason of doing the satellite mission in the first place, which is eventually sending the captured images to the Earth. Moreover, the domain does not account for the image quality at all. Images captured by satellites are sent to ground stations and used for multiple purposes, such as meteorology, oceanography, fishing, agriculture, geology, and so on. With the continuous advancements in satellite technologies and their imagery systems, it is now possible to take high-resolution images of the desired phenomena. However, this comes at the cost of having a large volume of data that has to be managed and sent to Earth. This promotes the use of on-board image compression systems to reduce data volume and makes it easier to send images to the Earth (Yu, Vladimirova, and Sweeting 2009; Muthukumaran and Ravi 2015). The compressed images are then decompressed at the receiving side. Image compression technologies in satellites can be either *lossy* or *lossless*. Lossless compression is used to reduce the size of images without losing information. On the other hand, lossy compression is used when a higher compression of images is needed, but with an acceptable loss of information. Multiple algorithms can be used to achieve each type of compression with varying compression and decompression speed and image quality. For example, CCSDS, Wavelet, Bandelet, and JPEG 2000 algorithms can be used for image compression (Indradjad et al. 2019). While the CCSDS and JPEG 2000 algorithms have two versions, one for lossy compression and another for lossless compression, the Wavelet and Bandelet algorithms can only be used for lossy compression.

Realistic domain aspects. The original Satellite domain does not account for sending captured images to Earth, and it does not consider the image quality and image compression. Extending the Satellite domain by giving the option of compressing the image before sending it to Earth and by giving the option of what compression algorithm to use addresses the following realistic domain aspects.

Aspect 1. The original Satellite domain is built initially from a real application of planning but with some simplifications. Extending the benchmark with these alternatives contributes more to the realism of this domain.

Aspect 2. The original domain already has tasks that address many specificities of this application, e.g., the preparation of the satellite to take images. However, adding the capability to send an image and alternatives brings the domain even closer to the real application.

Aspect 3. The extensions represent realistic alternatives that exist in this domain.

Aspect 7. As high-resolution images and compression methods are a result of the advancements made in satellite technologies, this extension makes the domain up to date with the recent technologies and trends in this domain.

Extension. The first extension of this domain is the addition of the task of sending a captured image to Earth. This is a compound task that can be decomposed by two methods. The first one entails compressing the image and then sending it to Earth, while the other method entails sending the image in its original size to Earth. We model these two methods as strict alternatives. The first method results in an action that sends the image to Earth. The second method, however, decomposes the task into a compound task of compressing and then sending the image. The compression task represents the second extension of this domain. This compound task can be further decomposed by two methods for compressing the image lossless/lossy and sending it to Earth. Each of these two methods results in a compound task that can be decomposed using multiple methods. Each method corresponds to a specific compression algorithm and refine the task into two actions; the first one compresses the image according to the chosen algorithm and the second action sends the image that was compressed using a specific method.

Encoding. The extension consists of 9 methods and 12 actions encoded in HDDL. Listing 1 shows the method for sending an image directly. The method requires four parameters related to the satellite, direction of this satellite, instrument that is used to take the image, and mode of the image. Its task network consists of one primitive task represented by the action shown in Listing 2. This action requires having the captured image and has the effect of sending the image, thus making it unavailable. The method for lossy compression of images is shown in Listing 3. To be applicable, the method requires having the image uncompressed, which is modelled by two predicates. The rest of methods and actions are modelled similarly.

Listing 1: Method for sending an image directly, without compression.

```
(:method m11_sendDirectly
 :parameters (?s - satellite
              ?d - direction
              ?i - instrument
              ?m - mode)
 :task (send_to_earth ?s ?d ?i ?m)
 :precondition ()
```

```
 :ordered-subtasks(and
       (t1 (send_directly ?s ?d ?m))))
```

Listing 2: Action for sending image directly, without compression.

```
(:action send_directly
 :parameters (?s - satellite
              ?d - direction
              ?m - mode)
 :precondition (and (have_image ?d ?m))
 :effect (and (image_sent ?s ?d ?m)
              (not(have_image ?d ?m))))
```

Listing 3: Method for lossy compression and sending of an image.

```
(:method m12_compress_image_send_lossy
 :parameters (?s - satellite
              ?d - direction
              ?i - instrument
              ?m - mode)
 :task (compress_image_send ?s ?d ?i ?m)
 :precondition (and (have_image ?d ?m)
           (not(image_compressed ?s ?d ?m)))
 :ordered-subtasks(and
      (t1 (compress_image_send_lossy
                               ?s ?d ?i ?m))))
```

5.2 Rover Domain

Application. Planning problems in the Rover domain deal with exploring a planet via multiple rovers (Ramoul et al. 2017; Pellier and Fiorino 2021). The rovers navigate through various locations, collect rock and soil samples, and take pictures of target objects. Getting rock or soil samples consists of navigating to the location of the data, emptying the store of the rover, taking the sample, and navigating to a location from which the lander is visible, and lastly, communicate the collected data to a lander. Getting images consists of camera calibration, navigating to a location from which the target is visible, capturing the image, and lastly, relaying the image to the lander by navigating to a location from which the lander is visible and transmitting the image to it.

Motivation. Rovers are designed to drive on certain types of ground, hence, rovers might not be able to traverse certain locations. With the advancements in technology, however, rovers' possible missions have increased. In particular, *space drones* have been introduced in planet exploration missions with the goal to help rovers, and perhaps astronauts in the future, to explore terrains that are hard to reach by rovers. Space drones are "small uncrewed vehicles guided by remote or autonomous means" (Carr et al. 2021). In July 2020, NASA sent the first space drone called *Ingenuity* (also known as Mars Helicopter) to Mars. This drone is attached to the rover called *Perseverance rover*, which landed safely on Mars in February 2021 (Potter 2020).

Realistic domain aspects. Extending the Rover domain by giving the option of rovers to get help from space drones when collecting and transferring data addresses the following realistic domain aspects.

Aspect 1. The original Rover domain was modelled on the basis of an actual application of planning technology, that is, the Mars exploratory rovers mission of 2003. The domain, however, no longer reflects exploratory rovers missions as many improvements have been made in terms of capabilities and technology. One such improvement is the integration of space drones as helpers for rovers. The extended Rover domain reflects this reality.

Aspect 2. The original domain already has tasks that represent specificities of this application, e.g., communicating rock data to the lander. Thus, it goes beyond restating tasks that are common in many domains, e.g., navigation tasks. Adding the space drones to help rovers, however, is a step forward in exploiting the application's specificities.

Aspect 3. Adding space drones as helpers of rovers allows modelling possible real-world alternatives for achieving space mission tasks. In particular, each task of communicating image, soil, or rock data to the lander in the domain can be decomposed using two alternative methods. The first method is for the rover to communicate the collected data to the lander by navigating towards it. The second method, however, gives the option to use the drone to communicate the data to the lander. Having these alternatives reflects better the wide spectrum of alternatives that can exist in real missions of rovers with drones.

Aspect 7. The extended Rover domain reflects some of the new aspects of the latest Mars exploration mission. In particular, it incorporates the use of space drones, which can be considered as transformative technology in rovers missions. Thus, the extended Rover domain does not only reflect current real applications, but it also opens a pathway for novel applications of planning.

Extension. The Rover domain can be extended to model a possible usage of space drones in planet exploration missions. The extension can include several alternatives: one alternative for sending collected data to a lander, another alternative to let a space drone collect data instead of a rover, and an alternative to let the collection of data be a collective work between rovers and drones. In our extension, we implemented the first alternative. However, implementing the other alternatives follows a similar model as the one we provided. Thus, the extended domain contains drones in different locations such that each space drone belongs to a rover. To send data to the lander using a space drone, the drone navigates to the rover's location, the rover transfers the data to the drone, the drone navigates to a location from which the lander is visible, and lastly, it communicates the data.

Encoding. As the original Rover domain does not allow recursive navigation between different locations, we modified the navigation task as proposed by the total-order version of the domain. Listing 4 shows the method that can be used to communicate soil data to the lander by the drone. The method decomposes the `send_soil_data` task into the following subtasks. The drone navigates to the rover's location, the rover transfers the data to the drone, the drone navigates to a location from which the lander is visible, and lastly, it communicates the data. This method requires that the drone belongs to the rover and the lander should be visible from the location to which the drone should navigate. Listing 5 shows the method used to decompose the task of navigating the drone to a desired location. This method is modelled similarly to the navigation method of the rover. In particular, it is a recursive method that decomposes `navigate_drone_abs` eventually into primitive tasks for navigating one step at a time between connected locations. However, unlike rovers that cannot traverse all locations, for a drone to move to a location it is sufficient that this location is visible from its current location. Listing 6 shows the action of communicating the soil data from a drone to a lander. This action requires that the location of the lander is visible from the drone's location and that the drone has the collected soil data. As an effect, the drone will not have the data anymore, the soil data is communicated, and the channel of the lander is free again.

Listing 4: Method for sending soil data by drone.

```
(:method m-send_soil_data_drone
 :parameters (?rover - rover
              ?waypoint ?x ?y ?z - waypoint
              ?l - lander
              ?drone - drone)
 :task (send_soil_data ?rover ?waypoint)
 :precondition (and (at_lander ?l ?y)
                (visible ?x ?y)
                (belongsTo ?drone ?rover)
                (at ?rover ?z))
 :ordered-subtasks (and
           (navigate_abs_drone ?drone ?z)
           (transfer_soil_data_to_drone
                ?rover ?drone ?waypoint)
           (navigate_abs_drone ?drone ?x)
           (communicate_soil_data_drone
             ?drone ?l ?waypoint ?x ?y)))
```

Listing 5: Recursive method for navigating the drone to the desired location.

```
(:method m-navigate_abs_drone-1
 :parameters (?drone - drone
              ?from ?to - waypoint)
 :task (navigate_abs_drone ?drone ?to)
 :precondition (at_drone ?drone ?from)
 :ordered-subtasks (and (visit ?from)
             (navigate_abs_drone-1
                       ?drone ?from ?to)
             (unvisit ?from)))
```

Listing 6: Action for communicating soil data from the drone to the lander.

```
(:action communicate_soil_data_drone
 :parameters (?r - drone
              ?l - lander
              ?p ?x ?y - waypoint)
 :precondition (and (at_drone ?r ?x)
             (at_lander ?l ?y)
             (have_soil_analysis_drone ?r ?p)
             (visible ?x ?y)
             (channel_free ?l))
 :effect (and (channel_free ?l)
              (communicated_soil_data ?p)
   (not (have_soil_analysis_drone ?r ?p))))
```

6 Discussion

Most of the benchmark HTN domains are inspired by real applications and provide structural diversity by considering specificities of the real-world domain. However, most domains do not consider alternative ways of achieving tasks, i.e., do not model alternative methods to decompose compound tasks. The lack of having choice of methods when decomposing a compound task does not mean that there are no other planning choices that planners can make to compute different plans. Current benchmark HTN domains allow planners to reason about possible combinations of methods, and subsequently, different combinations of operators, bindings, and ordering of partially ordered tasks. However, we point out and demonstrate that adding alternative methods to decompose a compound task widens the spectrum of possibilities to accomplish tasks. This, in turn, enables planners to make informed choices that are not solely enforced by the world state, potentially not only increasing the chance of finding plans but also leading to more refined plans. Our extensions move towards that direction and make a step forward towards realistic domains.

6.1 Quantification of the Two Domains

To gain more insights into the original and extended versions of the Satellite and Rover domains, we analyse their structural diversity using the structural metrics defined in Section 4. Table 2 shows a comparison between the original and extended version of each of the two domains considering the structural metrics. We highlight that both original domains do not offer alternatives for performing tasks. However, after the extension, the Satellite domain has 10 strict alternative methods that can achieve the task of compressing the image, whereas the Rover domain has 6 weak alternative methods that offer performing the transmission tasks to the lander by the drone. Since the domain is modelled with the assumption that all rovers have drones, these 6 weak alternative methods are guaranteed to offer the choice of performing the task. Thus, in this special case, these 6 methods can be considered strict alternative methods.

As a result of having more alternatives, the number of total methods in both domains increased. Moreover, the number of added alternative methods with 1, 2, or 3 subtasks in the Satellite domain increased after the extension. Similarly, the number of methods with 3 or 4 subtasks in the Rover domain increased. However, the maximum number of subtasks among all methods did not increase in both domains.

Outside of the scope of realistic domain aspects, we can gain further insights into the realism of domains by looking into their *knowledge richness*, which is a property enabled by HTN planning. Unfortunately, there is no accepted definition of what makes the knowledge in domains richer. We

Table 2: Comparison between the original and extended versions of Satellite and Rover domains.

Domain		Structural metrics										Domain-richness metrics				
		#m	**#ms**	**#mw**	**#mn**	**#t**	**#rt**	**#m-tn**				**#p**	**#pm**	**#ot**	**#otp**	**#a**
								1	**2**	**3**	**4**					
Satellite	**Original**	8	0	0	8	6	0	2	4	1	0	8	5	4	0	5
	Extended	17	10	0	7	6	0	6	9	2	0	10	7	4	0	17
Rover	**Original**	11	0	0	11	9	0	2	4	1	4	26	12	7	1	11
	Extended	17	0	6	11	12	1	2	4	2	8	31	17	8	1	18

could argue that the knowledge richness can also be quantified in terms of the number of tasks, methods, alternative methods, and expressiveness constructs. However, it is hard to objectively asses whether the addition of any of these constructs enriches the knowledge. Thus, we take a more pragmatic approach and quantify the knowledge richness using metrics that involve domain objects and relationships between them. The *domain-richness metrics* are: (1) number of predicates **#p**; (2) number of predicates that define the relation between two or more objects **#pm**; (3) number of children object types, which are the leaves in the types hierarchy **#ot**; (4) number of object types that are parent in the object types hierarchy **#otp**; and (5) number of actions **#a**.

Table 2 shows that the numbers of actions, predicates, and predicates that define the relation between two or more objects increased in both domains as a result of the extensions. Having more information about possible objects and their relations gives more knowledge about the domain. Similarly, the number of actions that can be performed in a specific domain gives a direct indication about what can be done in the domain. Thus, the knowledge richness also indicates that the extended HTN domains can be considered closer to reality.

6.2 Beyond the Two Domains

Having in mind the realistic domain aspects, we can follow the same approach to extend other benchmark HTN domains. For example, the *Elevator* domain, listed in Table 1, can be extended with at least four alternatives for automatic and manual control of the opening/closing of the doors when leaving/entering elevators. Extending the domain with tasks of opening/closing a door and modeling alternative ways of how to achieve them can contribute to the satisfaction of **Aspect 1**, **Aspect 2**, and **Aspect 3**. In particular, the option of manual control originates from a real need in emergency situations (Sharma et al. 2005), thus contributing to **Aspect 1**. Moreover, the extension allows expressing a wider range of ways to achieve the tasks of opening/closing doors (**Aspect 3**) and increases the domain's structural diversity – rather than just modelling the moving up/down tasks, which are similar to movement actions in other domains (**Aspect 2**).

A step further would be to generalise the results in Table 2. One can observe that it may not be necessary for HTN domains to contain both strict and weak alternatives. Some domains may be characterised by strict alternatives only, while others by weak ones only. Another observation is that adding strict or weak alternatives requires extending the domains with more building blocks, such as predicates and objects. This inevitably increases the domain richerness.

6.3 Quality-Oriented Planning Approaches

If there are no criteria based on which one evaluates alternatives, the choice of a method might be done either randomly or even by taking the first method of a task as encoded in the domain model, which is the case with some HTN planners (e.g., (J)SHOP2 (Ilghami 2006)). However, since different planning choices can lead to different plans with probably different quality, it is of utmost importance for planning approaches that guide the planning process towards computing quality plans, or what we call *quality-oriented approaches*, to make choices in an informed manner based on the quality users aims for. In fact, having a range of alternatives might be seen as a first step for creating domains for quality-oriented approaches that allows a fair comparison other than performance between the different approaches. This is because the availability of alternatives allows planners to make informed choices that are not merely enforced by the world state, which empowers planners to find and refine plans.

Consider a domain whose primitive tasks or actions have costs and we need to compute *cost-optimal plans*, that is, a plan that has the minimal sum of the costs of its constituent actions. Consider now the motivating example. The person might weigh the alternatives based on the comfort, time, or both as paid costs for each choice. In this case, s/he will make informed choices, which, in turn, will contribute to the quality of the trip plan. If we consider time, then each choice that minimises the time spent on roads to the destination will lead eventually to a trip with time-optimal quality.

Take now, for example, the Satellite domain, where the cost can be defined in terms of time, image accuracy, or any combination of them. In this case, a planning approach that is oriented toward computing quality plans should take into account, for example, that sending a compressed image is faster than sending an image without compression. However, there is also some time spent in compressing the image. This time depends further on the compression algorithm used. On the other hand, the accuracy of sending an image without compression or using lossless compression algorithm is definitely higher than using a lossy algorithm for compression. This demonstrates something that can be generally stated as follows: the way in which a quality-oriented planning approach weighs its choices depends on the quality they aim for, which in turn controls the final plan.

7 Conclusions

Benchmarks have a paramount effect on the quality and reliability of the tools they are used for. Planning is no exception and, in general, the community around it strongly relies on competitions and shared benchmarks to progress the state of the art. Here, we address the issue of dimensions for modelling benchmark HTN domains and improving them on several aspects, most notably, the modelling of alternative methods for tasks. We point out that to have HTN domains closer to actual applications, we need to increase the spectrum of choices of how to perform tasks according to real cases and considering the recent trends in the actual applications. To illustrate our idea, we extended two existing domains for which we also provide HDDL encodings.

We posit that the extended domains are advantageous for developing HTN planners of a broader applicability, bringing them, in turn, closer to applications. We plan to do so in our own research and to use them as a fair and meaningful means of comparing with planners of equivalent expressive power. We also plan to study formally the properties of these domains, their complexity, and explore the effects of the extensions on the complexity. Finally, we are interested in exploring the extension of other benchmark HTN domains while considering the remaining realistic domain aspects.

References

Behnke, G.; Höller, D.; Bercher, P.; Biundo, S.; Pellier, D.; Fiorino, H.; and Alford, R. 2019. Hierarchical planning in the IPC. In *Workshop on HTN Planning (HPlan)*, ICAPS.

Behnke, G.; Höller, D.; and Bercher, P., eds. 2021. *Proceedings of the 10th International Planning Competition: Planner and Domain Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC 2020).*

Bercher, P.; Alford, R.; and Höller, D. 2019. A Survey on Hierarchical Planning - One Abstract Idea, Many Concrete Realizations. In *International Joint Conference on Artificial Intelligence*, IJCAI, 6267–6275.

Carr, C.; Samnani, M.; Tani, J.; McKaig, J.; Hammons, E.; Newman, D. J.; Ho, K.; Ekblaw, A.; and Truelove, N. 2021. Space Drones: An Opportunity to Include, Engage, Accelerate, and Advance. In *Bulletin of the American Astronomical Society*, volume 53.

Currie, K., and Tate, A. 1991. O-Plan: The open planning architecture. *Artificial Intelligence* 52(1):49–86.

Cushing, W.; Weld, D. S.; Kambhampati, S.; Mausam, K. T.; and Talamadupula, K. 2007. Evaluating Temporal Planning Domains. In *International Conference on Automated Planning and Scheduling*, ICAPS, 105–112.

Erol, K.; Hendler, J. A.; and Nau, D. S. 1994. UMCP: A Sound and Complete Procedure for Hierarchical Task-network Planning. In *Artificial Intelligence Planning Systems Conference*, volume 94 of *AIPS*, 249–254.

Georgievski, I., and Aiello, M. 2015. HTN planning: Overview, comparison, and beyond. *Artificial Intelligence* 222:124–156.

Georgievski, I., and Aiello, M. 2016. Automated planning for ubiquitous computing. *ACM Computing Surveys* 49(4):1–46.

Hoffmann, J.; Edelkamp, S.; Thiébaux, S.; Englert, R.; Liporace, F.; and Trüg, S. 2006. Engineering Benchmarks for Planning: the Domains Used in the Deterministic Part of IPC-4. *Journal of Artificial Intelligence Research* 26:453–541.

Höller, D.; Bercher, P.; Behnke, G.; and Biundo, S. 2018. A generic method to guide HTN progression search with classical heuristics. In *International Conference on Automated Planning and Scheduling*, ICAPS.

Höller, D.; Behnke, G.; Bercher, P.; Biundo, S.; Fiorino, H.; Pellier, D.; and Alford, R. 2020. HDDL: An extension to PDDL for expressing hierarchical planning problems. In *AAAI Conference on Artificial Intelligence*, AAAI, 9883–9891.

Ilghami, O. 2006. Documentation for JSHOP2. Technical report, Department of Computer Science, University of Maryland.

Indradjad, A.; Nasution, A. S.; Gunawan, H.; and Widipaminto, A. 2019. A comparison of Satellite Image Compression methods in the Wavelet Domain. *IOP Conference Series: Earth and Environmental Science* 280(1):12–31.

Muthukumaran, N., and Ravi, R. 2015. The performances analysis of fast efficient lossless satellite image compression and decompression for wavelet based algorithm. *Wireless Personal Communications* 81(2):839–859.

Nau, D.; Cao, Y.; Lotem, A.; and Munoz-Avila, H. 1999. SHOP: Simple hierarchical ordered planner. In *International Joint Conference on Artificial Intelligence*, IJCAI, 968–973.

Nau, D. S.; Au, T.-C.; Ilghami, O.; Kuter, U.; Murdock, J. W.; Wu, D.; and Yaman, F. 2003. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research* 20:379–404.

Pellier, D., and Fiorino, H. 2021. From Classical to Hierarchical: benchmarks for the HTN Track of the International Planning Competition. Technical Report 2103.05481, arXiv.

Potter, N. 2020. A Mars helicopter preps for launch: The first drone to fly on another planet will hitch a ride on NASA's Perseverance rover. *IEEE Spectrum* 57(7):06–07.

Ramoul, A.; Pellier, D.; Fiorino, H.; and Pesty, S. 2017. Grounding of HTN planning domain. *International Journal on Artificial Intelligence Tools* 26(5):113–120.

Sacerdoti, E. D. 1975. The nonlinear nature of plans. In *International Joint Conference on Artificial Intelligence*, IJCAI, 206–214.

Sharma, T. S.; He, Y.; Odgers, C.; and Mahendran, M. 2005. Acceptable use of vertical transport systems during fire emergencies. In *Proceedings of Australian Structural Engineering Conference, 11-14 September 2005, Newcastle, Australia*, 40–49.

Wilkins, D. E. 1988. *Practical planning: Extending the classical AI planning paradigm*. Morgan Kaufmann Publishers Inc.

Yu, G.; Vladimirova, T.; and Sweeting, M. N. 2009. Image compression systems on board satellites. *Acta Astronautica* 64(9-10):988–1005.