Agent Goal Management Using Goal Operations

Sravya Kondrakunta [†] SRAVYA1@STOLAF.EDU
Venkatsampath Raja Gogineni[*] GOGINENI.14@WRIGHT.EDU
Michael T. Cox[*] MICHAEL.COX@WRIGHT.EDU
[†]Department of Mathematics, Statistics and Computer Science, St. Olaf College, Northfield, MN 55057 USA

[*]Department of Computer Science & Engineering, Wright State University, Dayton, OH 45431 USA

Abstract

Goal management for autonomous agents remains an unsolved problem in AI and cognitive systems research. At the highest level, an agent must be capable of performing various operations on its goals to solve a problem. For example, some goal operations include selection, change, formulation, delegation, and monitoring of goals. The literature includes several approaches to goal management with both implicit and explicit attention to goal operations. For example, some approaches perform a cost-benefit analysis to select the next goal to pursue when multiple goals exist. Although past work implements single and multi-goal operations, none of them directly shed light on an agent's response when multiple-goal operations are possible simultaneously. This paper develops an procedure to address agent goal-management when multiple operations co-occur and shows how managing the interaction between them improves performance.

1. Introduction

The area of *goal reasoning* focuses on the management of goals for intelligent agents and cognitive systems (Aha, 2018; Roberts et al., 2018). Goal management involves several individual operations on goals such as goal selection, goal change, goal formulation, goal monitoring, goal achievement, and goal delegation. Although some current research efforts explicitly implement various goal operations (Johnson et al., 2016; Kondrakunta, 2017, 2021), none of this work focuses on agent choices when it faces the possibility of multiple goal operations concurrently. For example, the decision to respond to a major impediment as opposed to the pursuit of a new opportunity that benefits the agent's mission is one such choice. In the former case, the agent can change the goal to a similar but achievable goal; whereas, in the latter instance, the agent can formulate a new goal to leverage the opportunity. How to balance that decision is an important open problem. Here, two goal operations present themselves, and the agent must either choose one or order one before the other. This paper presents a goal management procedure for these kinds of decisions and reports empirical evidence to support its efficacy.

Goal reasoning focuses on developing genuinely autonomous agents that are robust in complex, dynamically changing worlds. For us, autonomy implies that a cognitive system should not have to wait for a human to provide new goals when events go awry (Cox, 2013) or to determine that a given goal should be delegated to another agent better equipped to achieve it. To facilitate goal reasoning in general and our conceptualization of autonomy more specifically, we organize goal management into a set of separate but interacting goal operations (Cox, 2017). Each operation is designed to perform a unique function defined as the following.

- **Goal Selection:** Choose a current goal to pursue from an agenda of all the agent's goals (Kondrakunta, 2017; Kondrakunta & Cox, 2017; Kondrakunta et al., 2021).
- **Goal Change:** Change one goal to a different but similar goal such that the difference is minimal (Cox et al., 2017). Goals can undergo sequences of transformations (Cox & Dannenhauer, 2016; Cox & Veloso, 1998) including priority shifts (Choi, 2011) and in extreme cases, abandonment (Cox & Dannenhauer, 2017; Harland et al., 2017).
- **Goal Formulation:** Generate a new goal when the agent finds itself in an unexpected problem situation (Cox, 2007, 2013; Paisner et al., 2013).
- **Goal Monitoring:** Track whether a goal is still relevant or necessary (Dannenhauer & Cox, 2018).
- **Goal Delegation:** When an agent cannot achieve its goals, it can assign its own goal/goals to another agent (Gogineni et al., 2021).
- **Goal Achievement:** Execute plan steps to achieve a goal state and then determine that the world is actually in that state.

One can observe the work on goal operations implemented across several domains. In particular, Aha et al. (2013), Gogineni et al. (2019), and Weber et al. (2012) each present research on goal formulation where agents generate their own goals by reasoning about their situations and motivations. Also, Cox et al. (2017) outlines the importance of goal management, presents multiple-goal operations, and implements a small number of them. A recent article by Aha (2018) also broadly explains the importance of goal operations and outlines several key ideas and relevant domains. As mentioned, all the literature mentioned focus on individual goal operations as opposed to their co-occurrence.

The problem presented in this paper is, when multiple-goal operations co-occur, how can an agent choose the order for their execution. For example, the agent cannot continue working on a selected goal when an anomaly in the environment hinders the pursuit of that goal. In contrast, the agent can continue pursuing a selected goal when the anomaly does not pose a problem to its goals. In such situations, an agent must make an informed decision whether to continue its goal or to protect itself from harm. Therefore, we develop an approach that draws numerical inferences from observations. Converts the inferences to rules to match with a preexisting rule-base. This match aids the agent make an appropriate decision in aforementioned situations.

The remainder of the paper is as follows: Section 2 describes an automated goal management procedure to address the agent's response in case of co-occurrence of multiple goal operations. Section 3 introduces the marine and construction problem domains to demonstrate the application of the procedure. Section 4 presents the experimental setup and empirical results obtained in the two domains. Section 5 discusses related research. Finally, Section 6 presents the closing remarks and future research directions.

2. A Goal Management Procedure to Select Goal Operations

Goal operations can co-occur in multiple scenarios. One such scenario is an event of an anomaly, the agent can encounter anomalies that affect the agent negatively or positively. Negative anomalies require the agent to spend additional resources while delaying the mission. However, positive anomalies help the agent discover beneficial information about the world. The agent can use the information later to benefit its mission. This paper focuses on scenarios when the agent discovers an anomaly that affects the agent negatively. Also, the presented goal management procedure focuses mainly on goal selection, change, and formulation. But, the procedure is generic enough to include other goal operations. To demonstrate this, we provide an example of goal delegation (Kondrakunta, 2021). We implement the procedure in an open-source cognitive architecture called *Meta-cognitive Integrated Dual-cycle Architecture (MIDCA)*(Cox et al., 2016)

The procedure developed chooses one goal operation in case of multiple goal operations using the following three factors:

- **Anomaly Effects:** When in an anomalous situation, the agent must reason about the anomaly.
- **Goal Priority:** The agent must have a general idea of how important each goal is to the mission.
- **Resource Availability:** The agent must also consider the number of resources required for all the goals while deciding on goal operations.

The agent now makes use of the above three factors and reaches a decision for a goal operation to pursue. The following subsections detail the use of each of these in turn.

2.1 The Effects of an Anomaly on the Agent

An unexpected event in a dynamic world can affect the agent positively or negatively. As mentioned previously, this paper focuses only on negative anomalies. To study the negative effects of the anomaly, we further sub-categorize the effects into three categories: negative effects of the anomaly on agent's goals, negative effects of the anomaly on agent's health, and the number of times the anomaly repeats (which is not a direct negative effect, but it can estimate the cumulative negative effect of the anomaly in future). So far, we have identified three factors to study the negative effects of an anomaly on the agent. To estimate the negative effects the agent must have a model to approximate the values. After building models for each subcategory, we translate the outputs to qualitative factors. Currently, the qualitative factors we use are high and low. We present the importance of such qualitative classification in later sections.

Figure 1 shows the three factors. Specifically, the agent looks at the negative effects of an anomaly on the agent's goals, the negative effects of an anomaly on the agent's anomalies, and the anomaly repetition factor. The next subsections the modeling of each of the factors.

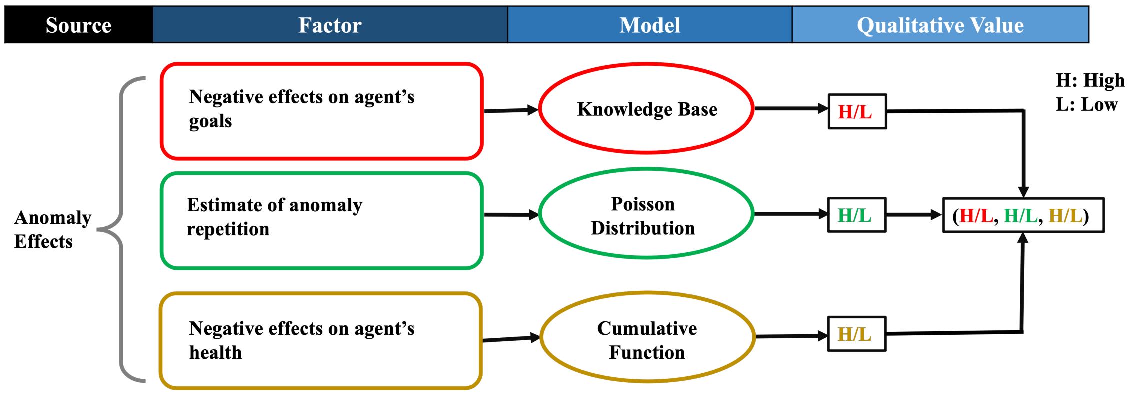

Figure 1. Effects of an anomaly on the agent. The figure represents three specific factors: the first is the negative effects on the goals of the agent; the second factor estimates the negative effects on the agent's health; the third factor estimates the repetition of the anomaly. In addition, the figure also presents models to estimate each of the factors. It also depicts the qualitative estimates obtained from the quantitative models.

2.1.1 Negative Effects on Agent's Goals

To model the negative effects of the anomaly on the agent's goals, we use a knowledge base. If the anomaly occurred previously, the agent finds a case relating to the negative effects in the knowledge base and retrieves the case. After retrieval, the agent then classifies the negative effects as high or low. If the agent encounters the anomaly for the first time, the agent considers the anomaly to be a higher risk anomaly. Later, based on its encounter, it creates a new case for the new anomaly and stores it in the knowledge base.

2.1.2 Negative Effects on Agent's Health

To model the negative effects of the anomaly on an agent's health, we created a cumulative numeric function 1 to track it. We assume that the agent's health starts at a maximum value of 100. The cumulative function keeps track of the agent's health by considering many sub-factors that affect health. For example, consider that n is the total number of sub-factors affecting the agent's health, and h_i is the numeric value of health affected for each factor. Then, we define the overall cumulative function as in equation (1).

$$H = \Sigma_{i=1}^{n}(h_i) \tag{1}$$

Every sub-factor affecting the agent's health is different. Hence, we can define each h_i using one of the basic functions: simple linear, quadratic, exponential, or logarithmic function. For example, the actions an agent performs to achieve its goal linearly reduce its health (over time, the battery of the agent reduces). Therefore, for one goal achieved, we define the factor by which the health decreases as $h_{goal\_number\_1} = x_1 * g_1$; where x is a value between 0 to 1, and g_1 is also a domain-specific value set by a human expert for each goal type. Of course, the exact value of x for each goal differs. Currently, a human expert provides the x value. So, for 'm' goals, the sub-factor (performing actions) would be a simple summation given in equation (2).

$$h_1 = h_{actions} = \Sigma_{y=1}^{m}(h_{goal\_number\_y}) = \Sigma_{y=1}^{m}(x_y * g_y) \tag{2}$$

Consider a second sub-factor of fire. Fire affects the agent's health exponentially. Hence, similar to a linear function, we should use an exponential function to model the negative effects of fire. Let us indicate the second sub-factor in the equation (3) where t is the time and e is Euler's number.

$$h_2 = h_{fire} = e^t \tag{3}$$

For example, let us consider the two sub-factors ($h_{actions}$, h_{fire}) defined and substitute them in the cumulative function 1. The sequence of steps below presents the calculation for the agent's health.

$$H = \Sigma_{i=1}^{2}(h_i) = h_1 + h_2$$
$$H = h_{actions} + h_{fire}$$
$$H = \Sigma_{y=1}^{m}(h_{goal\_number\_y}) + e^t$$
$$H = \Sigma_{y=1}^{m}(x * g_y) + e^t$$

We perform similar calculations for other sub-factors to include them in the cumulative function. We calculate the agent's health value before and after the anomaly. If the difference looks higher than an expert set threshold, we classify the health affected as high; otherwise, it remains low.

2.1.3 Anomaly Repetition

Finally, one of the last categories that we need to consider is to check the frequency of the anomaly repetition. Although the anomaly might not affect the agent's current goal or health, if it occurs very frequently, the agent should possess knowledge about the anomaly because it might be of interest to the agent in the future. Since the occurrence of an anomaly is an independent event, we use Poisson distribution to predict the frequency of occurrence of the anomaly by storing its past occurrences.

$$P(x) = \frac{e^{-\lambda}\lambda^x}{x!} \tag{4}$$

Equation (4) depicts the Poisson distribution function. Where λ is the mean of the anomaly occurrences over time, and x is the number of occurrences. The plot of the Poisson values obtained for several x values looks like a bell curve. We then determine the peak value of the bell curve to be the frequency of anomaly repetition. If the determined frequency value obtained exceeds a certain threshold set by an expert, the agent classifies the outcome as high else low.

2.2 Understanding the Importance of an Agent's Goals

The second major factor we used in the procedure to select goal operations is determining the priority of each agent's goal. Every agent needs to understand the importance of each goal it achieves or tries to achieve. To capture such a factor, we modeled the goal types of Schank & Abelson (1977). Here Schank broadly categorizes goals into several types and prioritizes one type over another. We used three goal types that best fit our goal management procedure out of all the goal types. Specifically, we consider *crisis:* goals generated in response to a crisis event; *preservation:* goals that help in the agent's self-preservation or resource preservation; and *achievement:* goals provided to the agent to achieve a task or reach a goal state. In general, we prioritize crisis goals over preservation and preservation over achievement goal types. We name these three factors qualitatively as C, P, A. C refers to a crisis, P refers to preservation, and A refers to achievement. Figure 2 presents the goal types in pictorial format.

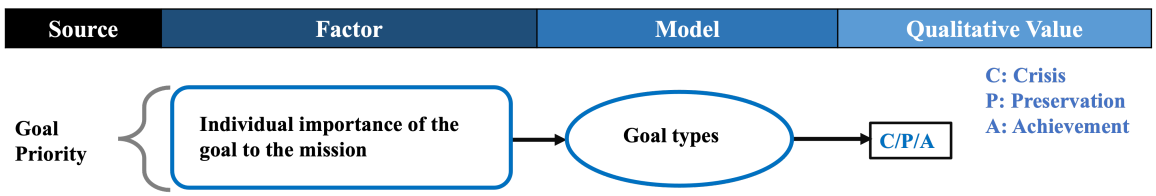

Figure 2. Goal priority defined by goal type. The agent estimates the priority of goals based on the goal type. Currently, there are three specific types of goals: Crisis, preservation, and achievement.

2.3 Keeping Track of the Agent's Resources

The third major factor we use in the procedure to choose goal operations is to continuously check the resource availability for goal achievement. To model this factor, we use a knapsack algorithm. Such an estimation algorithm is crucial because it can provide a rough estimate of the goals an agent could achieve with a lower cost compared to the planning cost for all the goals. A knapsack algorithm uses a resource value to pick the maximum set of goals an agent can achieve. If the agent predicts that the agent can achieve greater than 85% of its goals with the resources available, then it sends a qualitative value "Yes"; otherwise, it sends a qualitative value "No." Figure 3 depicts such in a pictorial format.

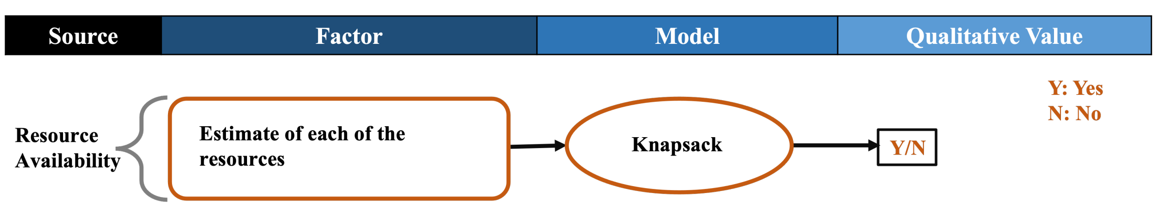

Figure 3. Resource availability for agents' goals. The agent estimates the amount of resources based by following a knapsack algorithm. The agent converts the numerical estimates from the knapsack to qualitative values with a threshold.

2.4 Choosing a Goal Operation

Now we have access to all the qualitative factors required to make an informed decision for selecting a goal operation. We use all three factors and formulate generic rules. The generic rules are instrumental across several domains. Figure 2 depicts all the components of the procedure together. Specifically, some of the rules we used to make a decision are as follows:

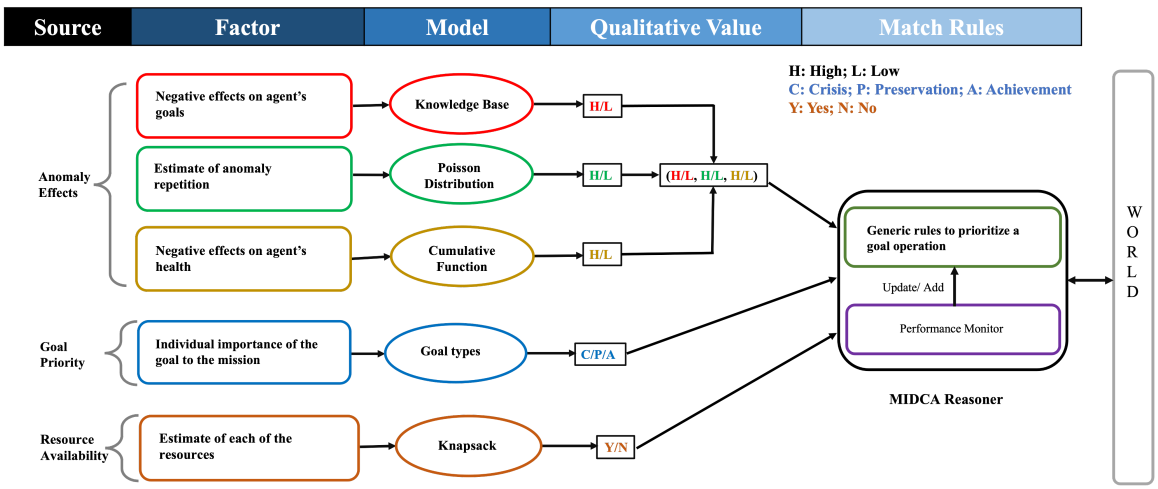

Figure 4. Flow chart for a goal management procedure to select goal operations. This figure depicts the combination of all estimates mentioned in figures [1][2][3]. In addition, it also displays the rule matching procedure in MIDCA(Cox et al., 2016).

After generating all the qualitative factors, the agent then compares the results to the generic rules in MIDCA. Each rule outputs a specific goal operation. If the factors match any of the rules, the agent chooses a specific goal operation. Specifically, some of the rules we used to decide are as follows. In addition, the agent analyzes the rules in the given order.

- If any of the anomaly effects is *"high"* and the resources available are sufficient, *"yes,"* then prioritize goal formulation.
- If all the anomaly effects are *"low"* and resources are sufficient, *"yes,"* for both selection and change and selection and change generate different goal types, then prioritize one goal operation based on the goal type, *"Crisis > Preservation > Achievement.*
- If all the anomaly effects are *"low"* and resources are sufficient, *"yes,"* for both selection and change and selection and change generate same goal types, then prioritize one goal operation that uses fewer resources.

The rules mentioned above are generally sufficient to make an informed decision when formulation, change, and selection operations co-occur. However, the procedure can also easily include other goal operations. For example, we could easily include a delegation rule as follows:

- If resources available are not sufficient, *"no,"* then choose goal delegation over all the remaining operations.

In addition, we also realize that these generic might need adaptation based on the agent's situation in the real world. Therefore, we plan to update the rules or add new rules in the future through the reinforcement obtained from the real world. We implement the goal management method in two different domains: the marine survey domain and the construction domain.

3. Implementation Domains

We implement the procedure in two domains. They are the marine survey domain and the construction domain. Both domains have an autonomous agent that attempts to achieve certain initial goals. In both of the domains, the agent faces multiple unexpected situations. In such situations, the agent faces several choices, and it needs to make a decision without the help of a human expert. Before we attempt to understand the importance of the procedure, it is important to understand the domains. The next two subsections elaborate on both. The third provides a detailed example in the first domain to clarify the context for the evaluation that follows.

3.1 The Marine Survey Domain

Consider the problem of time-limited surveys of marine environments with *autonomous underwater vehicles (AUV*s). Typical missions measure salinity, temperature, and pressure throughout the water column and can incorporate acoustic receivers to investigate key aspects of marine life. An important feature within a marine ecosystem is the presence of *hot-spots* or regions of high fish density. These areas and the aquatic pathways between them that fish transit represent areas of ecological sensitivity. Thus, discovering the location of major hot-spots, especially for endangered species, is an important application. However, many barriers exist in such environments that make mission success difficult. Sea creatures may attach themselves to platforms and slow progress. Tides and currents exist that also impede progress and obstacles may appear requiring course change. Finally, conditions may change, limiting the detection range of acoustic receivers (Edwards et al., 2020; McQuarrie et al., 2021).

Our research team regularly deploys AUVs such as Slocum gliders and custom robotic fish as part of coastal observing systems, for science-driven experiments, and testing and evaluation of new platforms. During missions, the platforms surface to communicate on regular schedules or in response to forced interrupts. AUV surveys make a valuable contribution to management efforts in Gray's Reef National Marine Sanctuary, located on the inner shelf of the South Atlantic Bight off the coast of Savannah, GA (see Figure 5). Gray's Reef contains fish tagged with transmitters that send an acoustic signal or 'ping' at a pre-determined frequency (5 minutes for short experiments, 30-180 minutes for long-term tracking) containing identifiers unique to that instrument, allowing researchers to classify detection's by source.

We implemented the marine survey domain (Kondrakunta et al., 2021) using an open-source simulator to test search techniques prior to actual deployment and to empirically evaluate the mechanisms discussed in this paper. The simulator is called *Mission Oriented Operating Suite (MOOS)*

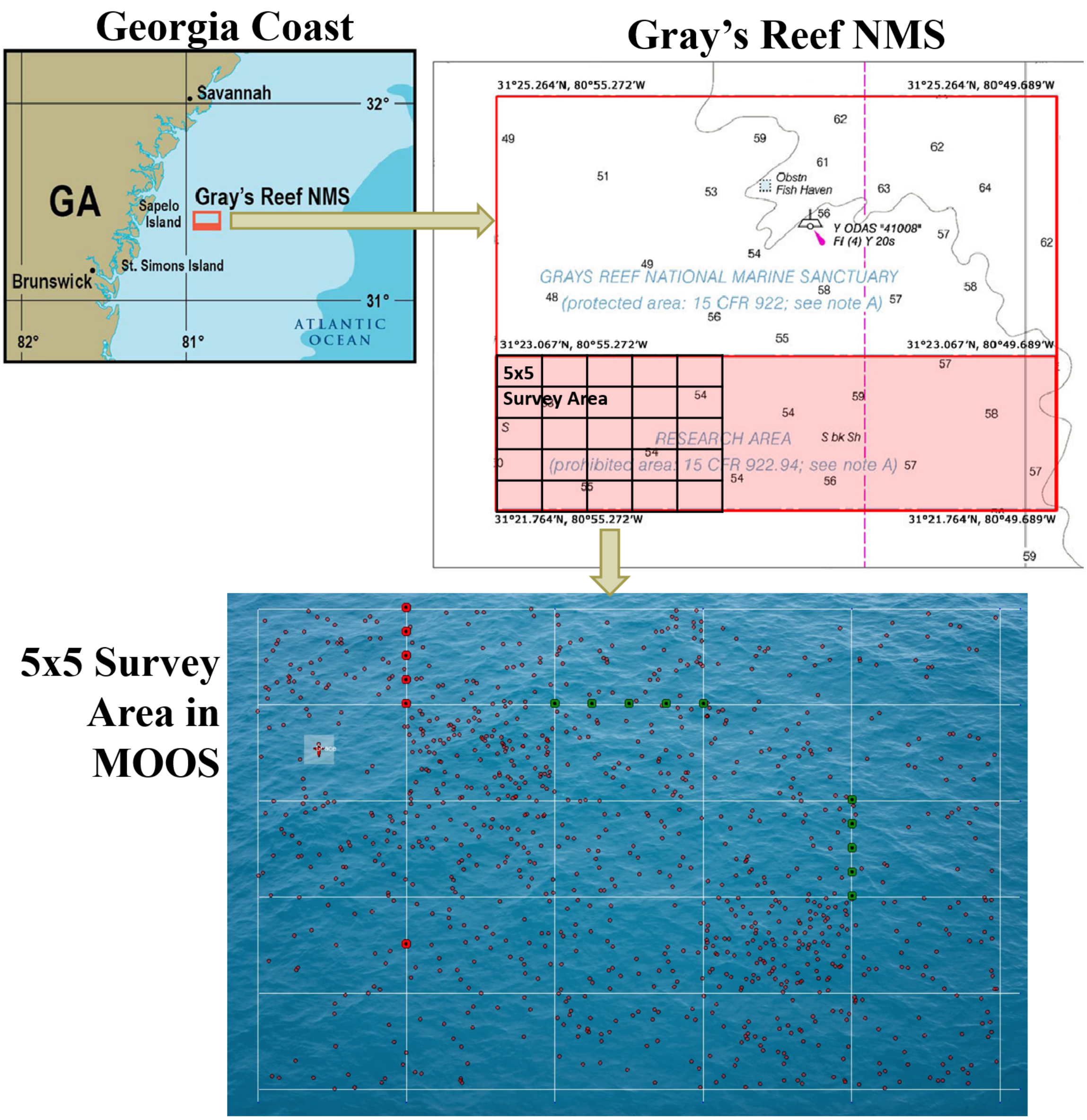

Figure 5. Gray's Reef National Marine Sanctuary is located off the coast of Georgia. It contains a research area off-limits to the public and indicated by pink coloring (see the lower portion of the right-hand expansion). Within the research area, we represent a 5x5 subsection having the origin in the lower left hand corner. Figure 5 expands this grid in the lower left corner. The sanctuary contains concentrations of fish (i.e., hot-spots) that are of interest to marine scientists. For example, the subsection shown here has hot-spots within cells (2,4) and (4,2). The agent is shown as a red cylinder in cell (1,4). The highlighted square around the agent signifies the sensor range for detecting acoustic fish tags (the small red dots).

(Benjamin et al., 2010), it provides autonomy for underwater platforms. The lower left of Figure 5 shows the portion of the research area modeled by the MOOS simulator and split into 25 cells. One scenario of fish distribution is shown in the lower left of the figure. The red dots depict 1000 fish (currently assumed to be static) that emit a ping every 17 time steps. In this cell, a hot-spot is located near the co-ordinate (2,4) and (4,2). The red streak represents a simulated AUV controlled by an agent, and the highlighted square area represents the receiver detection radius. At Gray's Reef, the detection radius varies with environmental conditions, but currently the simulator assumes it to be constant. As mentioned, an agent can identify hot-spots based on the number of pings.

3.2 The Construction Domain

The construction domain (Cox et al., 2017; Kondrakunta, 2017) is an extension of the classical blocks world domain (Fikes & Nilsson, 1971). Construction domain contains several blocks, and each block is given a unique name for identification. The domain contains two types of block: regular blocks and mortar blocks. Agent attempts to stack one block over the other to build high-raise towers. If the agent stack stacks the regular blocks the high-raise tower is wobbly. Whereas, if it uses mortar blocks the tower is sturdy. In general, sturdy towers are more desirable compared to wobbly towers. The agent gains reward for construction based on the type and height of the tower. There are two other actors in the domain due to which unexpected events in this domain arise. One is an arsonist who destroys the constructed towers by lighting them on fire. The second one is a thief who steals the construction blocks. Both the arsonist and thief act in random.

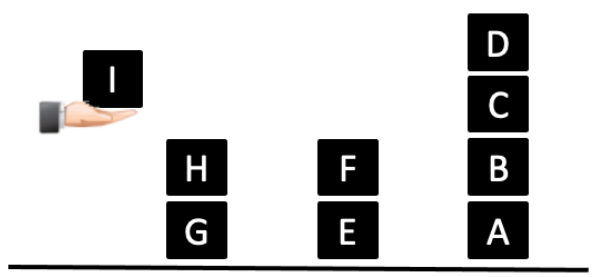

Figure 6. The figure depicts an instance of the achieved goals in the construction domain. The agent achieved three goals in the above scenario. The first tower has a height of four, the second tower has a height of two, and the third tower has a height of three.

Figure 6 shows an instance of the goals achieved by the agent in the construction domain. The agent constructs towers of different heights by placing the blocks on top of one another. Each goal provided to the agent is to construct one tower of a particular height. As mentioned, an agent gains a benefit after the successful achievement of the goal and also incurs a cost to achieve a goal. The benefit obtained for each tower is proportional to the height of the tower and its type. similarly,

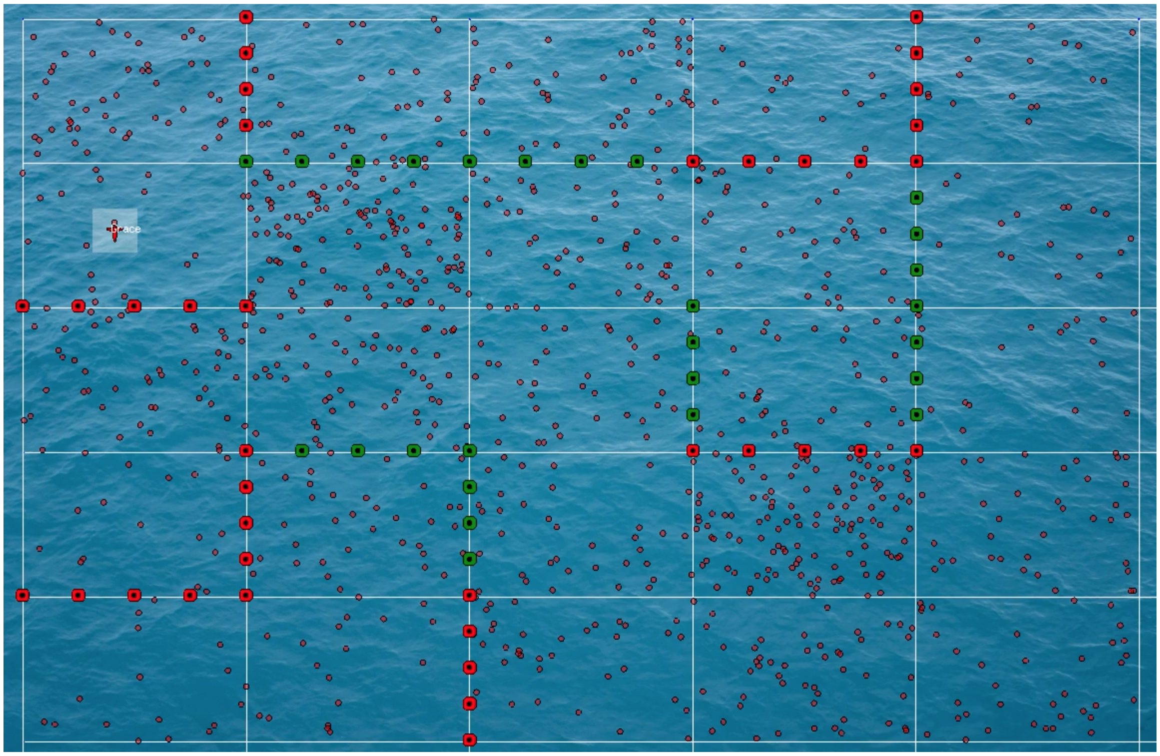

Figure 7. An example in the marine survey domain. The marine area is divided into 25 equal-sized cells. The anomalies in this domain are Remora attacks (cannot be depicted in the image) and Blockades (indicated by the red and green lines along cell edges). The agent (indicated by the red cylinder) is at the location (0,3). The highlighted area around the agent indicates the sensor range for detecting acoustic fish tags (the red dots).

the cost for constructing each tower is proportional to the number of blocks used for the tower. As mentioned, there is an arsonist that lights the constructed towers on fire. In addition, there is also a thief who steals the construction materials (or) blocks in random.

3.3 Working Example in the Marine Survey Domain

An example scenario from the marine survey domain can help us understand the importance of the agent using the proposed procedure to prioritize goal operations. Therefore, we refer to the agent using the proposed procedure as a smart agent.

Figure 7 represents an example scenario in the marine survey domain. As mentioned, there are twenty-five survey goals for the smart agent. We name each cell by its X, Y coordinate values, with both X and Y values ranging from 0-4. The initial location is the cell on the lower left (i.e., the origin whose coordinate is (0,0)). In addition, several types of anomalies exist in the domain. First, Remora attacks hinder the agent's movement; second, Blockades (represented in red and green lines) hinder the agent's movement from one location to the blocked location. The green blockades allow movement of the agent by either diving up or down. The agent can only learn about such a blockade if the agent stops and inspects the blockade to gain more knowledge. The red blockades do not allow the movement of the agent.

Consider a scenario where the smart agent performs goal selection and surveys the location (0,3). It then encounters a Remora attack. The agent has two choices; select a new survey location (Goal Selection), or formulate a goal to glide backward to respond to the Remora attack (Goal Formulation). The agent now reasons about the anomaly effects: Since the Remora attack hinders the agent's movement and the Remora also chips the paint off of the agent, the agent considers the anomaly to be a "high" threat to its health. According to the procedure outlined in Figure 4, if any anomaly effects are high, the agent must choose the goal formulation. So, as per the procedure, the smart agent prioritizes goal formulation and glides backward to free itself from the Remora attack. After which, it completes the survey in (0, 3). Since the agent does not have any current goal, it performs a goal selection operation and pursues a new goal. Let us assume that the agent selects the goal of surveying (0, 2). So, the agent must move from its current location (0, 3) to the destination location (0, 2).

The smart agent now encounters a blockade anomaly that hinders movement from (0, 3) to (0, 2). Thus, it now has two choices: select a new survey location between (1, 3) and (0, 4) or formulate a new goal to inspect the entrance to understand the cause of blockade. In this scenario, the blockade does not affect the agent's health or goals of the agent. In addition, since this is the first occurrence of blockade anomaly, the anomaly repetition is also low. Hence, the agent prioritizes goal selection and surveys the location given by the selection operation. Similarly, the smart agent prioritizes other goal operations as well. Such prioritization helps the agent address its goals promptly in a constantly changing world.

Section 4 will now illustrate the importance of carefully choosing goal operations when the domain is a dynamic one. We do this by implementing agents for multiple experimental conditions and comparing their performance in the two domains described above. The next section describes the experimental design and reports the empirical results obtained.

4. Experimental Design and Empirical Results

To demonstrate the effectiveness of the procedure we developed, we compare agent performance under four different experimental conditions. Out of the four, the first condition uses a baseline agent that performs only planning and execution. It uses no goal operations whatsoever. As such, the baseline condition represents an expected performance that blindly ignores all anomalies that arise and does not reason about goal choices. The second condition uses an ideal agent that acts in a world lacking any anomalies. The world is static and goal operations simply order the initial goals given to it. The third condition involves a random agent that chooses goal operations randomly when multiple operation are relevant. Finally, the fourth uses a smart agent that implements the decision procedure from Section 2 illustrated in Figure 4.

To evaluate performance empirically, we measure the average percentage of goals achieved as a function of time up until a given deadline. The goals of interest are the main goals of the domain, not ancillary goals formulated as a response to an anomaly or perturbation in the environment. Given the time constraint and the rate of impediments that arise, rarely can all goals be achieved in one trial. Hence, an agent must carefully choose which initial goals have priority to maximize its overall performance. This performance measure is similar to that used in over-subscription planning (García-Olaya et al., 2021; Smith, 2004; Speck & Katz, 2021). Goal reasoning research also uses similar measure to evaluate the agent performance (e.g., Dannenhauer et al. (2016)). Here we evaluate our approach in two very different domains. They are the marine survey domain and a tower construction domain.

4.1 Results for the Marine Survey Domain

In the marine survey domain, the agents search for hot-spots until reaching a deadline of 600 time units. The procedure of Section 2 improves the agents' robustness by responding to anomalies that affect the agent negatively as opposed to all anomalies. For example, remora attachments prompt goals to clear the remora from the vehicle. Control behavior that achieves such goals include flying backward; Whereas, agent ignores blockades unless they occur often.

The results are obtained with a multiple hot-spot scenario having hot-spots at locations (0,0), (0,4), (4,0), (4,4). A trial requires an agent to traverse the 25 cells of a scenario starting from a given initial location and find all hot-spots before the deadline. Each experiment has 100 trials randomizing their initial locations for each trial. We repeat the entire experiment with two other seed values. Therefore, we have 300 trials in total.

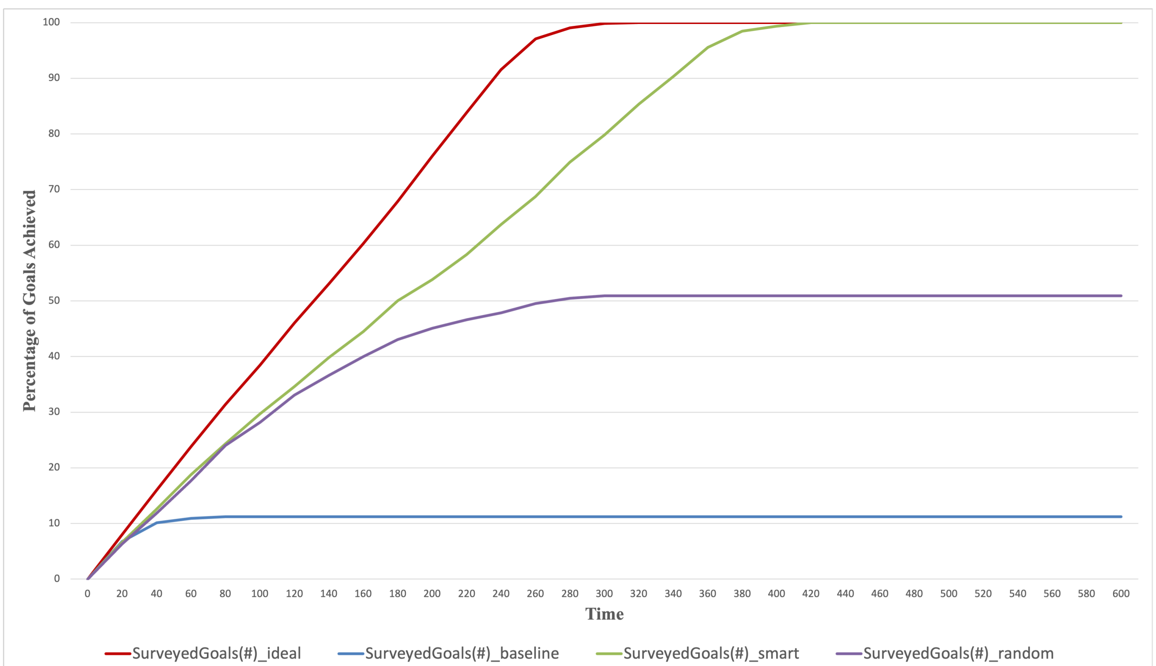

Figure 8. The figure depicts the results obtained in the marine survey domain. The X-axis denotes the time, and Y-axis represents the percentage of goals achieved. We compare the performance between the four different agent conditions: Ideal (agent working in a perfect world); Smart (agent using the proposed procedure); Random (agent selecting goal operations in random; and Baseline (the agent that ignores anomalies).

Figure 8 presents results for the strategies with and without the goal management procedure. In general, the results follow our expectation. Since the baseline agent ignores anomalies and only works towards goal achievement, it performs the worst. The ideal agent, which works without any anomalies in the domains performs the best. The agent choosing goal operations in random performs better than the baseline, but eventually it runs out of resources due to some poor choices. The smart agent, which uses the procedure to make an informed decision gradually matches the performance of an ideal agent.

4.2 Results for the Construction Domain

We used the same four experimental conditions mentioned in the marine survey domain for comparison. In construction domain, the agents search builds towers until reaching a deadline of 100 time units. Anomalies such as arsonist fires and theft occurs in random in this domain. The agent must either generate a goal to apprehend the criminal culprit or continue working on its current set of goals.

We provide the agent with multiple construction goals to achieve. A single trial consists of construction goals to build five to ten towers with differing heights from one to seven. We perform 2000 trials for each agent condition.

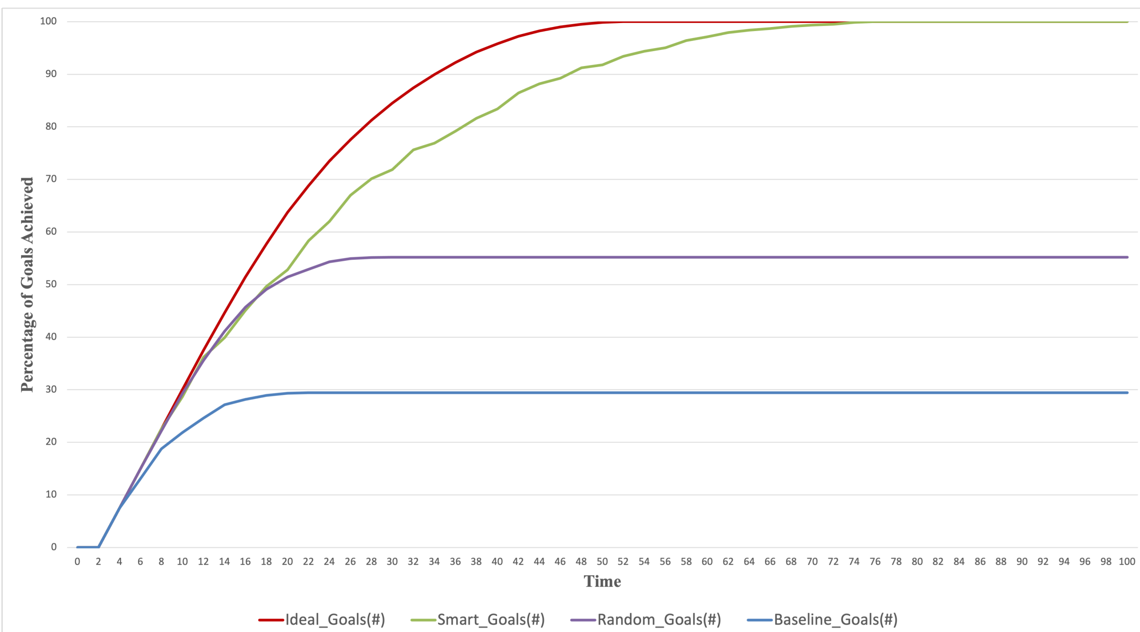

Figure 9. The figure depicts the results obtained in the construction domain. The X-axis denotes the time, and Y-axis represents the percentage of goals achieved. We compare the performance among four different agents: Ideal (agent working in a perfect world), Smart (agent using the proposed procedure), Random (agent selecting goal operations in random; and Baseline (the agent that ignores anomalies).

Figure 9 presents results for the strategies with and without the goal management procedure. In general, the results follow our expectation. Since the baseline agent ignores anomalies and only works towards goal achievement, it performs the worst. The ideal agent, which works without any anomalies in the domains performs the best. The agent choosing goal operations in random performs better than the baseline, but eventually it runs out of resources due to some poor choices. The smart agent, which uses the procedure to make an informed decision gradually matches the performance of an ideal agent.

5. Related research

Similar research effort in underwater platforms that also uses goal reasoning is Wilson et al. (2013a,b). The work formulates goals in unexpected situations using social, opportunity and exploration motivators. It then re-prioritizes the agent's goals based on the new goals. It also extends the work to multi-agent scenarios. Although Wilson shares common interests with the current paper, he avoids the issue of interactions between multiple goal operations. Also, Nelson & Schoenecker (2018) uses goal reasoning to improve a sonar sensor's performance, but avoids the interactions.

Apart from underwater platforms, similar goal-based autonomous behaviour is desirable and applicable in several other types of domains including space. For example, Troesch et al. (2020) presents the performance improvement in *Arcsecond Space Telescope Enabling Research in Astrophysics (ASTERIA)* CubeSat. ASTERIA was deployed into space to demonstrate precision photometry in 2017. In a similar domain, new goals are triggered based on the outcomes of previous goals (Chien et al., 2005). One contrast between both the domains discussed so far (underwater and outer space) are: most applications in underwater domains are massively under-determined, whereas many planning domains for outer space are over-subscription problems (Smith, 2004), requiring a fundamentally different approach. Although, they performs some goal operations that implicitly look like formulation, they do not focus on goal operations or their interactions. In general, goal operations are of interest to the goal reasoning community.

Goal reasoning more generally is relatively new area of research compared to many technical areas of AI, but they are active in the cognitive systems community. The applications of goal reasoning include its usage in the *Autonomous Response to Unexpected Events (ARTUE)*, (Klenk et al., 2013) system. The authors implement discrepancy detection and studies the performance variation of the ARTUE system in terms of benefits and limitations. The work on ARTUE is also further developed into several later versions called Trainable-ARTUE (T-ARTUE) Powell et al. (2011), and M-ARTUE Wilson et al. (2013b). T-ARTUE performs interactive learning to obtain knowledge to select goals. In addition, T-ARTUE also accepts criticism for a wrongly selected goal and corrects the selection not to repeat it in the future. M-ARTUE formulates goals based on domain-independent heuristics called motivators (opportunity, exploration, and social), where each motivator is weighed based on urgency and fitness. However, the above works also do not address the scenarios where multiple goal operations co-occur. Shivashankar et al. (2014) outlines solutions to a number of goal reasoning challenges. This work uses a hierarchical goal network structure to decompose a higher-level goal into several sub-goals to overcome particular limitations using *Motivated ARTUE (M-ARTUE)*. ARTUE and MIDCA agents both employ goal reasoning to identify and respond to unexpected situations in a dynamic environment, both use goal reasoning to perform several operations on their goals. Some of the works mentioned here might not use the goal reasoning explicitly, but they share similar goal-based behaviors. Hawes (2011) presents the motivation behind generation and selection of goals using motivational management framework. This work also surveys several other works on formalization and selection. However, it does not present any works that tackle co-occurrence of multiple operations. Gogineni et al. (2019) presents goal formulation through explanation patterns and anomaly detection. As mentioned, goal reasoning facilitates the implementation of various goal operations through a flexible array of methods to improve autonomy.

One reason for the applicability of goal reasoning in such diverse environments is due to its ability to perform various operations on its goals. For example, Rabideau et al. (2011) (inspired by Chien et al. (2005)) defines constraints and priorities to determine which goal to select among the set of all goals. Furthermore, domain-specific information metrics (e.g., distance traveled and time to perform cost estimation) can aid in goal selection (Johnson et al., 2016). This work is adapted and generalized to some domains using cost-benefit analysis (Kondrakunta & Cox, 2021). Trainable-ARTUE Powell et al. (2011) also presents goal selection with expert-based interactive learning. If the system selects a wrong goal, it accepts a penalty.

While the methods presented above take advantage of implementing various goal operations, none of them explicitly shed light on the agent's performance when there is a possibility for interaction of multiple goal operations. This paper addresses the issue by developing a method to prioritize one goal operation over another given the situation in two different domains. The uncertainty in both domains arise from the agent's limited communication when underwater, the unpredictable currents, the fish attacks, or due to external actors such as arsonist and thief. The next section concludes the paper and presents future research.

6. Conclusions and Future Research

In this paper, we explored the structure of goal management process when multiple-goal operations co-occur. Specifically, as opposed to two operations (Kondrakunta et al., 2021), we focus on three individual goal operations: selection, change, and formulation. This paper examines such problems with a focus on negative anomalous situations. We implemented a smart agent to handle the aforementioned scenarios. For comparison, we tested the smart agent against a baseline, an ideal, and a random agent. The results concluded that the importance and need of such a decision making agent.

Although, the results demonstrate the need for such an agent, the current agent still decides among a subset of goal operations. Therefore, we intend to extend this research in the future to include several other goal operations (e.g., goal delegation). An agent performs goal delegation in a multi-agent scenario, for an agent to pass its goals to a different agent. Specifically, goal delegation is particularly useful when the agent drifts off the survey region due to flow conditions, or in case of physical damage. It is smart for the agent to delegate its goals to a different agent to complete the mission. Having said that, we mentioned how we could integrate the goal delegation and others operations into the decision making process in earlier sections.

Acknowledgements

The National Science Foundation supported this research under grant 1849131 and the Office of Naval Research under grant N00014-18-1-2009. We also thank the reviewers for their many concrete suggestions.

References

Aha, D. W. (2018). Goal reasoning: Foundations, emerging applications, and prospects. *AI Magazine*, *39(2)*, 3–24.

Aha, D. W., Cox, M. T., & Muñoz-Avila, H. (2013). *Goal reasoning: Papers from the acs workshop (technical report cs-tr-5029)*. Technical report, University of Maryland, Department of Computer Science, College Park, MD.

Benjamin, M. R., Schmidt, H., Newman, P. M., & Leonard, J. J. (2010). Nested autonomy for unmanned marine vehicles with MOOS-IvP. *Journal of Field Robotics*, *27(6)*, 834–875.

Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davis, A., Mandl, D., Trout, B., Shulman, S., & Boyer, D. (2005). Using autonomy flight software to improve science return on earth observing one. *Journal of Aerospace Computing, Information, and Communication*, *2(4)*, 196–216.

Choi, D. (2011). Reactive goal management in a cognitive architecture. *Cognitive Systems Research*, *12*, 293–308.

Cox, M. T. (2007). Perpetual self-aware cognitive agents. *AI Magazine*, *28(1)*, 32–45.

Cox, M. T. (2013). Goal-driven autonomy and question-based problem recognition. *Second Annual Conference on Advances in Cognitive Systems* (pp. 29–45). Palo Alto, CA: Cognitive Systems Foundation.

Cox, M. T. (2017). A model of planning, action, and interpretation with goal reasoning. *Advances in Cognitive Systems*, *5*, 57–76.

Cox, M. T., Alavi, Z., Dannenhauer, D., Eyorokon, V., Munoz-Avila, H., & Perlis, D. (2016). MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. *Proceedings of the 30th AAAI Conference on Artificial Intelligence* (pp. 3712–3718). Menlo Park, CA: AAAI Press.

Cox, M. T., & Dannenhauer, D. (2016). Goal transformation and goal reasoning. *Proceedings of the 4th Workshop on Goal Reasoning*. New York, NY: IJCAI-16.

Cox, M. T., Dannenhauer, D., & Kondrakunta, S. (2017). Goal operations for cognitive systems. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 4385–4391). Menlo Park, CA: AAAI Press.

Cox, M. T., & Dannenhauer, Z. A. (2017). Perceptual goal monitors for cognitive agents in changing environments. *Proceedings of the Fifth Annual Conference on Advances in Cognitive Systems, Poster Collection*. Palo Alto, CA: Cognitive Systems Foundation.

Cox, M. T., & Veloso, M. M. (1998). Goal transformations in continuous planning. *Proceedings of the 1998 AAAI Fall Symposium on Distributed Continual Planning* (pp. 23–30). Menlo Park, CA: AAAI Press.

Dannenhauer, D., Munoz-Avila, H., & Cox, M. T. (2016). Informed expectations to guide GDA agents in partially observable environments. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (pp. 2493–2499). New York, NY: AAAI Press.

Dannenhauer, Z., & Cox, M. T. (2018). Rationale-based perceptual monitors. *AI Communications*, *31*, 197–212.

Edwards, C., Cho, S., Zhang, F., & Fangman, S. (2020). *Field and numerical studies to assess performance of acoustic telemetry collected by autonomous mobile platforms*. Technical report, National Marine Sanctuaries Conservation Series, Silver Spring, MD.

Fikes, R. E., & Nilsson, N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Proceedings of the 2nd International Joint Conference on Artificial Intelligence* (p. 608–620). San Francisco, CA, USA: Morgan Kaufmann.

García-Olaya, A., de la Rosa, T., & Borrajo, D. (2021). Selecting goals in oversubscription planning using relaxed plans. *Artificial Intelligence*, *291*, 103414.

Gogineni, V. R., Kondrakunta, S., Brown, D., Molineaux, M., & Cox, M. T. (2019). Probabilistic selection of case-based explanations in an underwater mine clearance domain. *Proceedings of the twenty seventh ICCBR Conference* (pp. 110–124). Otzenhausen, Germany: Springer.

Gogineni, V. R., Kondrakunta, S., & Cox, M. T. (2021). Multi-agent goal delegation. *Proceedings of the 9th Goal Reasoning Workshop*.

Harland, J., Morley, D. N., Thangarajah, J., & Yorke-Smith, N. (2017). Aborting, suspending, and resuming goals and plans in bdi agents. *Autonomous Agents and Multi-Agent Systems*, *31*, 288–331.

Hawes, N. (2011). A survey of motivation frameworks for intelligent systems. *Artificial Intelligence*, *175(5-6)*, 1020–1036.

Johnson, B., Roberts, M., Apker, T., & Aha, D. W. (2016). Goal reasoning with informative expectations. *Proceedings of the 4th Annual Conference on Advances in Cognitive Systems* (pp. 12: 1–14). Evanston, IL: Cognitive Systems Foundation.

Klenk, M., Molineaux, M., & Aha, D. W. (2013). Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence*, *29(2)*, 187–206.

Kondrakunta, S. (2017). *Implementation and evaluation of goal selection in a cognitive architecture*. Master's thesis, Wright State University.

Kondrakunta, S. (2021). *Complex interactions between multiple goal operations in agent goal management*. Doctoral dissertation, Department of Computer Science & Engineering, Wright State University, Dayton, OH.

Kondrakunta, S., & Cox, M. T. (2017). Autonomous goal selection operations for agent-based architectures. *Working Notes of the 2017 IJCAI Goal Reasoning Workshop*. Melbourne, Australia: University of Maryland, Department of Computer Science.

Kondrakunta, S., & Cox, M. T. (2021). Autonomous goal selection operations for agent-based architectures. *Proceedings from Advances in Artificial Intelligence and Applied Cognitive Computing*. Las Vegas, NV: Springer.

Kondrakunta, S., Gogineni, V. R., Cox, M. T., Coleman, D., Tan, X., Lin, T., Hou, M., Zhang, F., McQuarrie, F., & Edwards, C. R. (2021). The rational selection of goal operations and the integration of search strategies with goal-driven autonomy. *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*. Cognitive Systems Foundation.

McQuarrie, F., Woodson, C. B., & Edwards, C. (2021). Modeling acoustic telemetry detection ranges in a shallow coastal environment. *Proceedings of the 15th ACM International Conference on Underwater Networks and Systems (WUWNet'21)* (pp. 1–5). Guangdong, China: ACM.

Nelson, J. K., & Schoenecker, S. (2018). Goal driven autonomy as a model for reasoning in cognitive active sonar. *Proceedings of the 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)* (pp. 1–5). Sheffield, United Kingdom: IEEE.

Paisner, M., Maynord, M., Cox, M. T., & Perlis, D. (2013). Goal-driven autonomy in dynamic environments. *Goal Reasoning: Papers from the ACS workshop* (pp. 79–94). College Park, MD: University of Maryland, Department of Computer Science.

Powell, J., Molineaux, M., & Aha, D. W. (2011). Active and interactive discovery of goal selection knowledge. *Proceedings of the Twenty-Fourth International FLAIRS Conference* (pp. 413–418). Palm Beach, FL: AAAI Press.

Rabideau, G., Chien, S., & McLaren, D. (2011). Tractable goal selection for embedded systems with oversubscribed resources. *Journal of Aerospace Computing, Information, and Communication*, *8(5)*, 151–169.

Roberts, M., Borrajo, D., Cox, M., & Yorke-Smith, N. (2018). Special issue on goal reasoning. *AI Communications*, *31(2)*, 115–116.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Mahwah, NJ: Lawrence Erlbaum Associates.

Shivashankar, V., Kaipa, K. N., Nau, D. S., & Gupta, S. K. (2014). Towards integrating hierarchical goal networks and motion planners to support planning for human-robot teams. *AAAI Fall Symposium Series* (pp. 139 – 141). Arlington, VA: AAAI Press.

Smith, D. E. (2004). Choosing objectives in over-subscription planning. *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling* (pp. 393 – 401). Whistler, British Columbia, Canada: AAAI Press.

Speck, D., & Katz, M. (2021). Symbolic search for oversubscription planning. *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 11972–11980). Menlo Park, CA: AAAI.

Troesch, M., Mirza, F., Hughes, K., Rothstein-Dowden, A., Bocchino, R., Donner, A., Feather, M., Smith, B., Fesq, L., Barker, B., & Campuzano, B. (2020). Mexec: An onboard integrated planning and execution approach for spacecraft commanding. *Workshop on Integrated Execution (IntEx)/Goal Reasoning (GR), International Conference on Automated Planning and Scheduling (ICAPS IntEx/GR 2020)*.

Weber, B. G., Mateas, M., & Jhala, A. (2012). Learning from demonstration for goal-driven autonomy. *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence* (pp. 1176–1182). Toronto, Ontario, Canada: AAAI Press.

Wilson, M., Auslander, B., Johnson, B., Apker, T., McMahon, J., & Aha, D. W. (2013a). *Towards applying goal autonomy for vehicle control*. Technical report, Naval Research Lab, Navy Center for Applied Research in Artificial Intelligence, Washington, DC.

Wilson, M. A., Molineaux, M., & Aha, D. W. (2013b). Domain-independent heuristics for goal formulation. *Proceedings of the Twenty-Sixth International FLAIRS Conference* (pp. 160–165). St. Pete Beach, FL: AAAI Press.