Proceedings of the Ninth Goal Reasoning Workshop Submitted 10/2021; published 11/2021

Agent Goal Management using Goal Operations

Sravya Kondrakunta KONDRAKUNTA.2@WRIGHT.EDU
Venkatsampath Raja Gogineni GOGINENI.14@WRIGHT.EDU
Michael T. Cox MICHAEL.COX@WRIGHT.EDU
Department of Computer Science & Engineering, Wright State University, Dayton, OH 45431 USA

Abstract

Goal management in autonomous agents has been a problem of interest for a long time. Multiple goal operations are required to solve an agent goal management problem. For example, some goal operations include selection, change, formulation, delegation, monitoring. Many researchers from different fields developed several solution approaches with an implicit or explicit focus on goal operations. For example, some solution approaches include scheduling the agents' goals, performing cost-benefit analysis to select/organize goals, agent goal formulation in unexpected situations. However, none of them explicitly shed light on the agents' response when multiple-goal operations occur simultaneously. This paper develops an algorithm to address agent goal management when multiple-goal operations co-occur and presents how such an interaction would improve agent goal management in different domains.

1. Introduction

An agent goal management comprises several individual operations. For example, some goal operations include goal selection, goal change, goal formulation, goal monitoring, goal achievement, goal delegation. Although some of the research works (Johnson et al., 2016; Cox et al., 2017; Kondrakunta, 2017) explicitly implement the goal operations. None of the mentioned works focus on agent decisions when it faces a choice to respond to a minor anomaly (goal formulation) or pursue a new opportunity that benefits the agents' mission (goal change). In this scenario, two-goal operations co-occur, and the agent needs to choose one. This paper presents an algorithm for agents' decisions when multiple-goal operations occur at the same time. Also, the field of goal reasoning explicitly focuses on such goal operations for goal management.

Goal reasoning focuses on developing an autonomous agent which facilitates the implementation of various goal operations through a flexible array of methods to improve autonomy. Each of the goal operations performs a unique function defined as the following.

- Goal selection: choose a current goal to pursue from all of the agents' goals.
- Goal change: change a goal to a different goal such that the goal's objective remains the same where the approach might differ.

- Goal formulation: generate a new goal when the agent finds itself in an unexpected problem situation.
- Goal monitoring: surveil if a goal is still relevant to the agents' mission.
- Goal delegation: When an agent cannot achieve its goals, it can assign its own goal/goals to another agent.
- Goal achievement: plan and execute actions to achieve a goal state.

One can observe the work on goal operations implemented across several domains. For example, Aha et al. (2013); Gogineni et al. (2019); Weber et al. (2012) 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 some of them. A recent article by Aha (2018) also broadly explains the importance of goal operations and outlines several key ideas and relevant domains. The problem presented in this paper is when multiple-goal operations co-occur. For example, the agent cannot continue working on a selected goal when an anomaly in the environment hinders the agent from pursuing its goals.

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 a rule-based approach to tackle such situations.

The remainder of the paper is as follows: Section 2 describes the rule-based approach 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 algorithm. 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. Algorithm to Select Goal Operations

Goal operations can co-occur in multiple scenarios. The paper focuses on scenarios when the agent discovers an anomaly, which affects the agent negatively. Also, the algorithm presented focuses mainly on goal selection, goal change, and goal formulation. But, the algorithm is generic enough to include other goal operations, to demonstrate which we provide an example with goal delegation. We implement the algorithm in an open-source cognitive architecture called *Meta-cognitive Integrated Dual-cycle Architecture (MIDCA)*(Cox et al., 2016)

The algorithm 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 on a goal operation to pursue. The following few subsections dive deep into each of the factors mentioned.

2.1 Affects of an Anomaly on the Agent:

An anomalous(unexpected) situation encountered in a dynamic world can affect the agent positively or negatively. As mentioned previously, the paper focuses only on the negative anomalies. To study the negative effects of the anomaly, we further classify the negative effects into three categories: negative effects of the anomaly on agents' goals, negative effects of the anomaly on agents' health, and the number of times the anomaly repeats. So far, we have established that we need these three factors to study the negative effects of an anomaly on the agent. In addition, we use two parameters, high and low, to classify the outputs of each factor. We present the importance of such qualitative classification later. Now, we shift our gaze toward modeling each of the categories.

To model the negative effects of the anomaly on agents' 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.

To model the negative effects of the anomaly on agents' health, we assume that the agent's health is at a maximum value of 100. After which, we define a cumulative function. The cumulative function keeps track of agents' health for every item that affects their health. We define the cumulative function as:

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

Mathematical representation in (1) defines that n is the total number of factors affecting the agents' health, and h_i is the amount of health involved for each factor. Each h_i could be a simple linear, quadratic, exponential, or logarithmic function. For example, achieving goals requires movement, which linearly reduces health. Therefore, for one goal achieved, we define the factor by which the health decreases as $h_{i1} = x * g_{a1}$; where x is a value between 0 to 1. Of course, the exact value of x for each goal differs. A human expert provides the value, or the agent predicts from its experiences. So, for 'm' goals, the factor would be a simple summation given in (2).

$$h_i = \Sigma_{y=1}^{m}(h_{iy}) \tag{2}$$

We perform similar calculations for other non-linear factors. One example of an anomaly that might affect the agent's health exponentially would be an unexpected fire hazard. When an anomaly occurs, we have an actual health value from before the anomaly. We calculate agents' health after the anomaly. If the difference looks higher than an expert set threshold, we classify the health affected as high; otherwise, it remains low.

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 agents' 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.

$$P(x) = \frac{e^{-\lambda}\lambda^{x}}{x!} \tag{3}$$

Equation (3) depicts the Poisson distribution function. Where λ is the mean of the anomaly occurrences over time, and x is the number of desired occurrences. We plot the Poisson values obtained for several x values. The plot 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.

Figure 1 summarizes the negative effects of the anomaly on the agent. It presents the three anomaly effects we considered: negative effects on agents' goals, health, and anomaly repetition estimate. It also depicts the qualitative classification of the output, high or low.

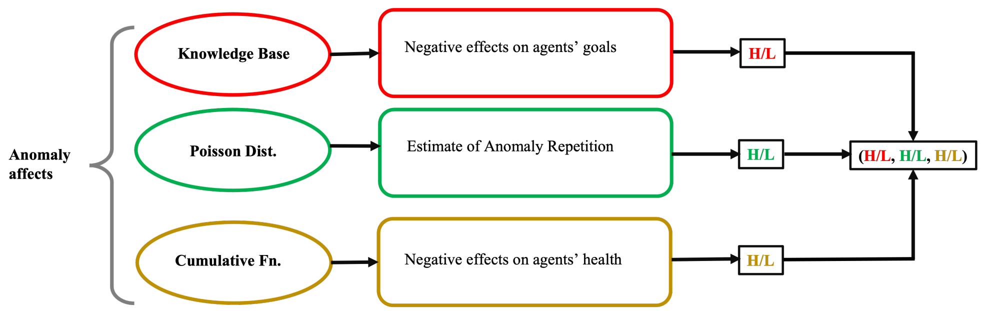

Figure 1. Affects of anomaly on the agent

2.2 Understanding the Importance of Agent's Goals:

The second major factor we used in the algorithm 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 model such a factor, we implemented the goal types defined by Schank & Abelson (2013). In Schank & Abelson (2013), he broadly categorized goals into several types and prioritized each goal type over the other. We used three goal types that better fit our decision algorithm out of all the goal types presented. We use three specific goal types, crisis: goals generated in response to a crisis/ anomalous situation; preservation: goals that help in agents' self or resource preservation; and achievement: the goals provided to the agent to achieve a task or reach a goal state. In general, prioritize we crisis goals over preservation and preservation over achievement goal types. Figure 2 presents the goal types in pictorial format.

2.3 Keeping Track of Agent's Resources:

The third major factor we use in the algorithm to choose goal operations is to check the resource availability for goal achievement continuously. To model such a factor, we use a knapsack al-

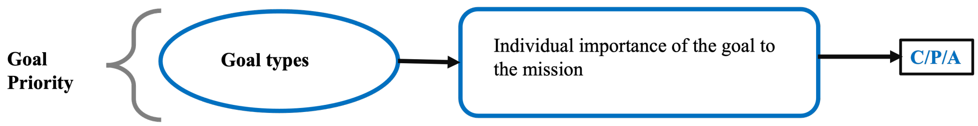

Figure 2. Goal priority defined by goal type

gorithm. 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.

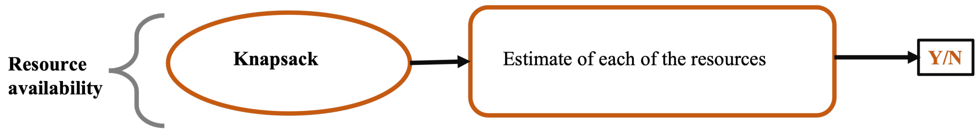

Figure 3. Resource availability for agents' goals

2.4 Choosing a Goal Operation:

We have access to all the qualitative factors required to make an informed decision on 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 algorithm together. Specifically, some of the rules we used to make a decision are as follows:

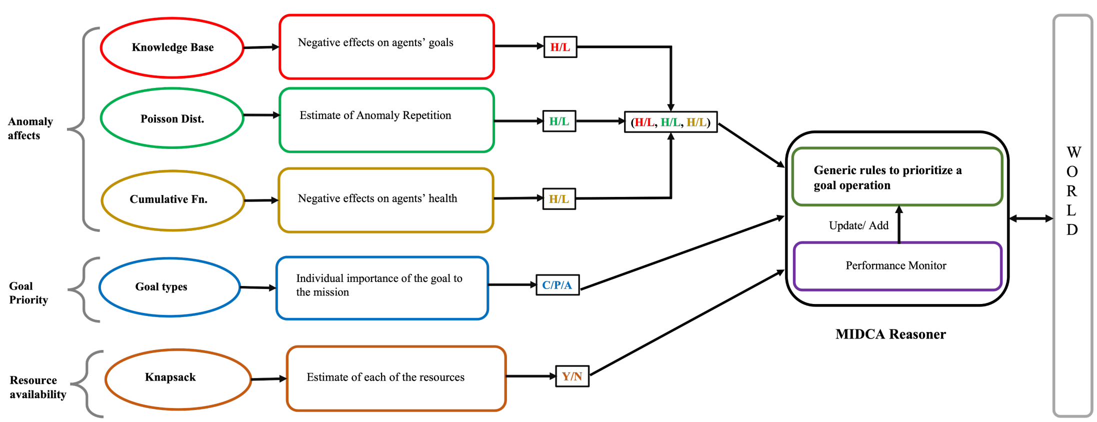

Figure 4. Algorithm to select goal operations

- If any of the anomaly affects is *"high"* and the resources available are sufficient, *"Yes"*, then prioritize goal formulation.

- If all the anomaly affects 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 affects 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.

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

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

Of course, we also realize that these generic might need adaptation based on several factors in the real world. Therefore, we update the rules or add new rules. Currently, we modify the rules through the reinforcement obtained from the real world. The algorithm is an extension to our previous work on goal selection and goal formulation Kondrakunta et al. (2021).

3. Implementation Domains

We implement the algorithm in two domains: Marine Survey Domain and Construction Domain. The two domain mentioned have an autonomous agent attempting to achieve certain goals. In both the domains the agents face 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 algorithm, it is important to understand the domains. The next subsections elaborate on both domains.

3.1 The Marine Survey Domain

Consider the problem of time-limited surveys of marine environments with *autonomous underwater vehicles (AUVs)*. 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. (Under Review).

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.

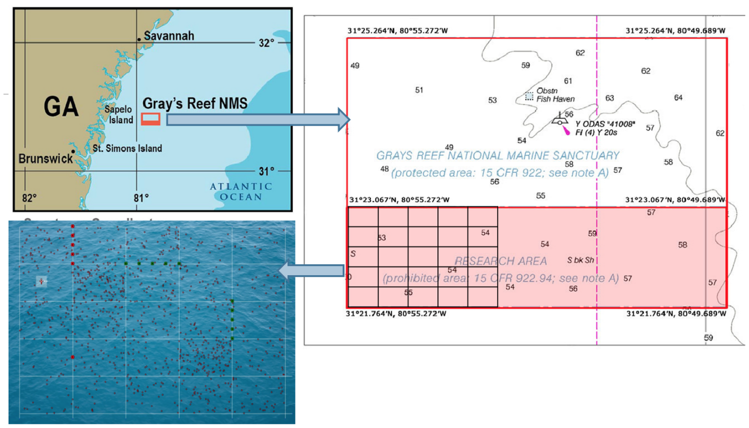

Figure 5. Gray's Reef National Marine Sanctuary is located off the coast of Georgia and contains a research area shown in the insert shaded in pink. Within this, we represent a 5x5 subsection. This grid contains fish hot-spots that are of interest to marine scientists, here within the cell at location (2,4) and (4,2). The agent (indicated by the red streak) is in (1,4) cell. The highlighted square around the agent indicates the sensor range for detecting acoustic fish tags (the small red dots).

We implemented the domain 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)*(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 is an extension of the blocks world domain. 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, 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.

The 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, 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.

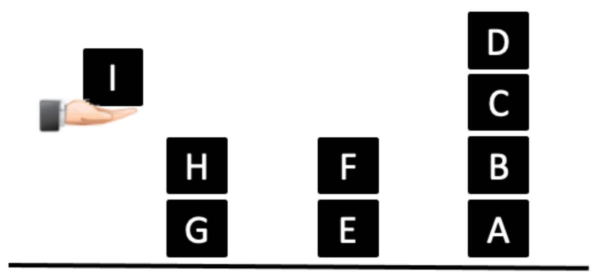

Figure 6. The figure depicts an instance of the achieved goals in the construction domain. The agent achieved three goals in the above scenario. The first tower has a height of four, the second tower has a height of two, and the third tower has a height of three.

Section 4 demonstrate the importance of choosing the goal operations by implementing and comparing the agents' performance in both the domains. It describes the experimental setup and empirical results obtained in both domains.

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 algorithm to prioritize goal operations. Therefore, we refer to the agent using the proposed algorithm 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 (its coordinate

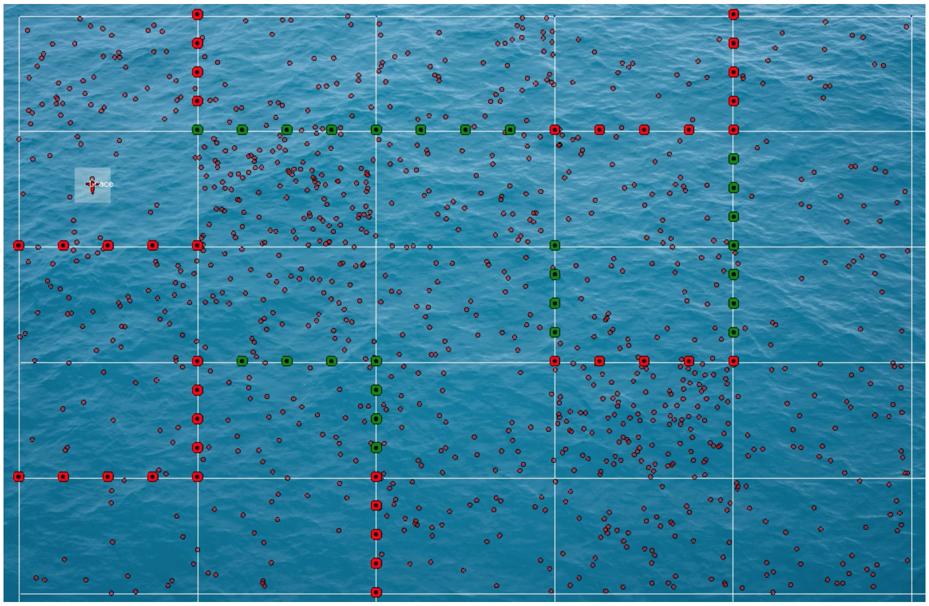

Figure 7. An example in the Marine Survey Domain, it is divided into 25 equal-sized cells. The anomalies in this domain are Remora attacks (cannot be depicted in the image) and Blockades (indicated by the red and green lines along cell edges). The agent (indicated by the red cylinder) is at the location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

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 algorithm 4, if any anomaly effects are high, the agent must choose the goal formulation. So, as per algorithm 4, 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 the dynamic world.

4. Experimental Design and Empirical Results

4.1 Results for the Marine Survey Domain

In order to demonstrate the importance of the algorithm we developed. We compared the performance of four different agents. Out of the four agents, the first is a baseline agent that performs only planning and no goal operations. Second is an ideal agent, it is the agent which acts in a world without any anomalies but with goal operations. Third is a random agent, it chooses a goal operation in random when multiple co-occur. Finally, the fourth is a smart agent, it implements the algorithm to choose a goal operation in case of co-occurrence.

In marine survey domain, the agents search for hot-spots until reaching a deadline of 600 time units. The algorithm 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 quite often.

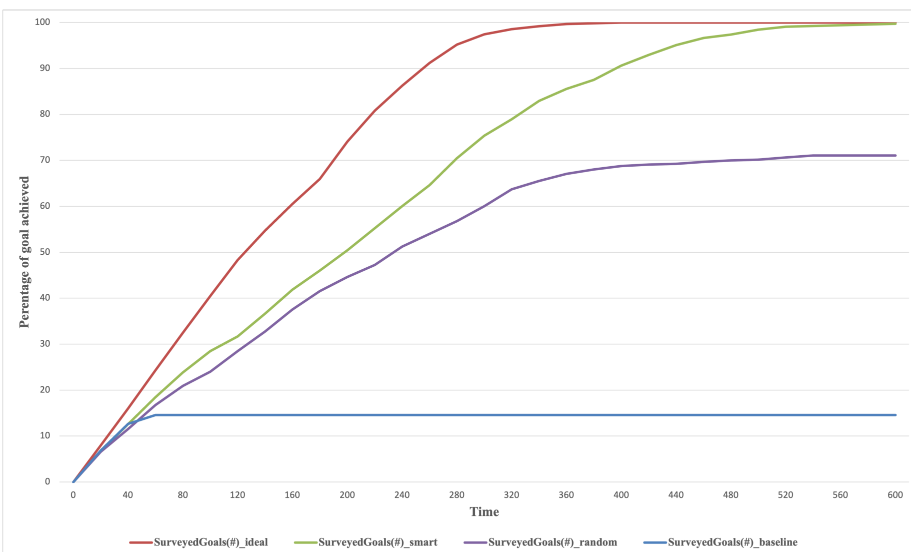

Figure 8. The figure depicts the results obtained in Marine Survey Domain. The X-axis denotes the time, and Y-axis represents the percentage of goals achieved. We compare the performance among four different agents: Ideal (agent working in a perfect world), Smart (agent using the proposed algorithm),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

We created a hot-spot pattern in the 5x5 survey region, where there is one hot-spot at the cell location of (1,4). 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. With 100 initial locations for each scenario, an agent repeats the experiment with two other seed values. Therefore, we have 300 trials in total.

Figure 8 presents results for the strategies with and without the algorithm. 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 algorithm 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 agents 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 fire and theft occurs in random in this domain. The agent must either generate a goal to pursue the culprit or continue working on its goals.

We provide the agent with multiple construction goals to achieve. A single trial consists of five to ten towers with different heights of one to five. We perform 30 trials for each agent.

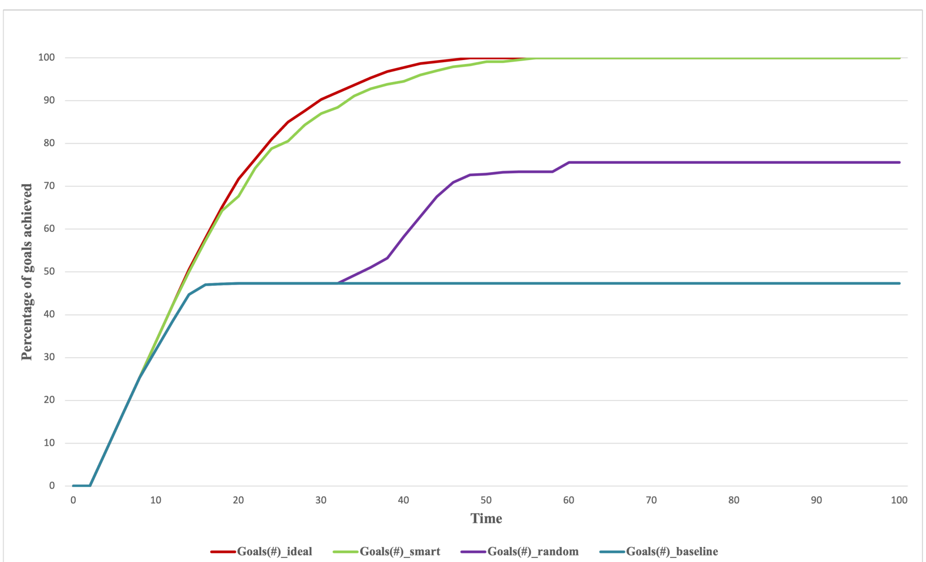

Figure 9. The figure depicts the results obtained in the Construction Domain. The X-axis denotes the time, and Y-axis represents the percentage of goals achieved. We compare the performance among four different agents: Ideal (agent working in a perfect world), Smart (agent using the proposed algorithm), 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 algorithm. 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 uneven curve for the random agent is because of its poor choices initially, but when it chooses a better goal operation it performs better before running out of resources. The smart agent, which uses the algorithm 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. 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. Gogineni et al. (2019) presents goal formulation through explanation patterns and anomaly detection. Goal formulation based on domain-independent heuristics called motivators (opportunity, exploration, and social), where each motivator is weighed based on urgency and fitness, are presented in M-ARTUE (Wilson et al., 2013b). 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 decision process when multiple-goal operations co-occur by extending pur previous work Kondrakunta et al. (2021). This paper examines such problems with focus of negative anomalous situations. For comparison, we presented the performance of the developed agent against a baseline, an ideal, and a random agent. The results concluded that the agent importance of such a decision making agent. The current agent decides among a subset of goal operations.

Therefore, we intend to extend this research to include several other goal operations (e.g., goal delegation) in the future. 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 grants 1849131, and also by the Office of Naval Research under grant N00014-18-1-2009. We also thank the reviewers for their 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.

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). AAAI Press.

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). San Francisco, CA: AAAI Press.

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.

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.

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., & Cox, M. T. (2021). Autonomous goal selection operations for agent-based architectures. *Proceedings from Advances in Artificial Intelligence and Applied Cognitive Computing* (p. in press). 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 ofsearch strategies with goal-driven autonomy. *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems* (p. in press). Cognitive Systems Foundation.

McQuarrie, F., Woodson, C. B., & Edwards, C. (Under Review). Modeling acoustic telemetry detection ranges in a shallow coastal environment. *Proceedings of the 14th ACM International Conference on Underwater Networks and Systems (WUWNet'21)*. 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.

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.

Schank, R. C., & Abelson, R. P. (2013). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

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.

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.