Guiding the Ass with Goal Motivation Weights

Hector Muñoz-Avila HEM4@LEHIGH.EDU
Computing Science and Engineering, Lehigh University, Bethlehem, PA 18015 USA

Mark A. Wilson MARK.WILSON@NRL.NAVY.MIL
David W. Aha DAVID.AHA@NRL.NAVY.MIL
Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, D.C. 20375 USA

Abstract

Goal reasoning (GR) is the study of free agents; they can autonomously and dynamically deliberate on and select what goals/objectives to pursue (Cox, 2007; Muñoz-Avila et al., 2010; Klenk et al., 2013; Vattam et al., 2013; Roberts et al., 2014). Endowing agents with this capability is particularly appropriate when the domain in which they operate is complex (e.g., partially observable, dynamic, multiagent), preventing the anticipation of all possible states and the precise pre-encoding of contingent plans for those states. Most GR agents monitor and assess the current state with respect to potential expectation violations or motivation triggers (Coddington et al., 2005). This deliberation may result in selecting an alternative to the goal(s) currently being pursued, requiring a planner to generate a corresponding set of actions for a controller to schedule and execute. In this paper, we study domain-independent goal selection, extending a method that combines motivations which was implemented in the GR agent M-ARTUE (Wilson et al., 2013). In particular, we relax the assumption that all motivations contribute equally to goal selection, and investigate the relationship between domain properties and motivator contribution in a paradigmatic domain. We view this as a step towards a deeper understanding of how motivations affect agent performance. Future work includes automatically learning motivator weights.

1. Introduction

In complex (e.g., partially observable, dynamic, and multiagent) environments, an autonomous agent may need to alter its own goals to be successful. For instance, an agent that flies an unmanned aerial vehicle may need to change its goal to refuel or recharge if it encounters unexpectedly strong headwinds. We refer to agents that can deliberatively select their own goals as *goal reasoning* (GR) agents. A crucial problem in GR is that of *goal selection* (i.e., deciding which goal(s) the agent should choose to pursue when it is appropriate to pick a new goal). One possible approach to goal selection is the use of domain-independent *motivators*, which encode high-level drives and rely on the agent's own internal models, as used by the M-ARTUE GR agent (Wilson et al., 2013). We refer to agents that employ such motivators as examples of *motivated agents*. M-ARTUE selects its goals according to a function defined on the following set of motivators:

- *Social Motivator*: This chooses user-provided goals.
- *Exploration Motivator*: This chooses goals that best expand the agent's world knowledge.
- *Opportunity Motivator*: This chooses goals that that maximize the agent's opportunity to act during plan execution, such as by conserving resources.

One way for an agent to combine these motivators is for it to assign the same weight to each motivator and repeatedly: (1) achieve some of the user's goals, (2) perform some exploration (from time to time), and (3) take action to conserve resources as necessary. However, this strategy might not work well in many situations.

For example, suppose an agent's user-provided goals involve delivering packages in a graph-like world where locations are nodes and edges are direct connections between locations with associated traversal costs. Suppose also that agents consume gas proportionally to these costs, gasoline stations are available in some (but not all) locations, and some information (e.g., some connections and the location of some gasoline stations) is initially unknown to the agent. In extreme situations, as in the Buridan's ass paradox,[1] the agent might oscillate between the three motivators, achieving few of the goals and (literally) running out of gas. Indeed, we hypothesize at least three scenarios where the balanced strategy might be inadequate:

1. *Observable Environment*: Most of the information is known to the agent. That is, most of the connections and gasoline station locations are known. In this case, performing exploration is not advisable as resources will be consumed for likely little benefit. Instead the agent should follow the **social** motivation to achieve the maximum number of user-provided goals, and use **opportunity** motivation to conserve resources otherwise. In this case no weight should be given to exploration.
2. *Hidden Environment*: Most of the information is not known by the agent. That is, the agent only knows about a fraction of the connections and gasoline station locations. Then the agent should emphasize **exploration** and place less emphasis on the Social and Opportunity Motivators until sufficient information has been gathered to fulfill most or at least many of the user-provided goals.
3. *High Resource Capacity*: The agent can retain a large quantity of resources that may be spent to achieve goals. In this case, it should emphasize the **Opportunity** Motivator and gather as many resources as possible for achieving social goals.

Figure 1 illustrates these three boundary cases. This raises the question of what kind of weight relations will exist among these motivators for the intermediate cases.

[1] The Buridan's ass (Rescher, 1959) is a philosophical paradox of an animal that is very thirsty and very hungry and can't decide between drinking from a nearby water fountain and eating from a nearby stack of hay that is located in the opposite direction from the fountain. The animal dies of hunger and thirst, never able to make a decision.

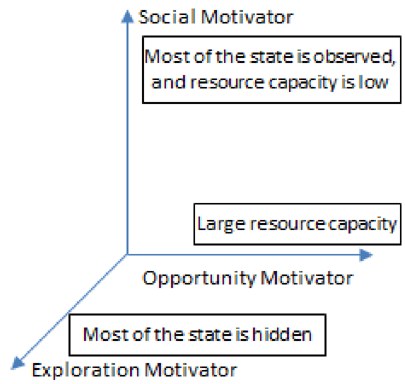


Figure 1. Boundary Cases for three Motivators

In this paper, we describe preliminary work on strategies for assigning different weights to the motivators in an M-ARTUE agent. Specifically, we study the relation between motivator weights and agent performance in different contexts, with the future objective of automatically learning weight settings that denote the relative, scenario-specific importance of the motivators. We present results from an initial investigation on the interaction of certain domain properties with the effects of motivator weight mixtures on agent performance. After summarizing related work (Section 2) and reviewing the M-ARTUE agent (Section 3), we report on a study (Section 4) using the Metric Transportation domain, a new variant of the Logistics Transportation domain (Veloso, 1994) whose characteristics were outlined above and are detailed further in Section 4. In our study, we found support that a motivated agent's performance varies when its motivators are given different weights, and that an agent's performance varies predictably based on domain and problem properties. We conclude and discuss future work in Section 5.

2. Related Work

Standard planning techniques can be used to solve problems in which all state information is known (Ghallab et al., 2004). In this situation, the planner can be tasked with achieving all package-delivery goals. The planner will attempt to achieve all the goals and backtrack as needed. If the planner is complete, it will generate one plan to achieve all of the goals, or indicate that such a plan doesn't exist. This can be interpreted as an instance of the motivators where all weight is given to the Social Motivator and no weight is given to the Exploration and the Opportunity Motivators.

Planning research has relaxed this "all-or-none" requirement for achieving the goals. Oversubscription planning attempts to find the maximum number of goals that can be achieved for a particular problem (Smith, 2004). Techniques such as expanding a planning graph, a compact representation of a set of solution plans, to select a subset of the goals to achieve has been explored for this purpose (Do et al., 2007). This can also be interpreted as an instance where all weight is given to the Social Motivator.

Our research is also related to planning for information gathering. In this form of planning, the agent navigates a partially observable state. Frequently, the fact that information must be gathered is explicitly expressed in a planning language as attached conditions to the actions (Draper et al., 1994) or as subgoals (Pryor & Collins, 1996). As a result, plans are generated that perform information-gathering tasks. This is akin to giving significant weight to the Exploration and Social Motivators in our work. A major difference with our work is that, in planning for information gathering, the amount of exploration will depend on the static encoding of the symbolic representation of the domain. In contrast, we are interested in the behavior of agents as weights for exploration and social activities, along with other behavior, are independently varied.

Domains such as the Metric Transportation domain have been a focus of optimal planning, which is the generation of plans that minimize or maximize some metric such as minimizing gasoline use (Williamson & Hanks, 1994). Optimal planning has been subject to extensive research (Kuffner, 2004) including using Djikstra-like search procedures that guarantee that the optimal solution will not be missed but at a potentially high computational cost of carrying out an exhaustive search in the plan space. As a result, optimal algorithms are computationally much less efficient than their non-optimal counterparts. Furthermore, partial observability has not been studied in the context of plan optimality, whereas in our work it is a central topic.

Our work is also related to replanning (Stentz, 1995), where a plan is modified as a result of changes in the environment. Techniques for replanning include using heuristics to determine the best way to complete a plan from the state where the change was detected. Most of the work on replanning concentrates on failures (e.g., the plan expected to find gasoline at a location but upon arrival it finds none). In contrast, we are considering an Exploration Motivator, which can be viewed as "exploring for the sake of exploring" even in situations where the current goals could be achieved. Our objective is to develop robust systems in which, for example, even when the goals change (e.g., new packages must be picked up and delivered) the system has pro-actively gathered information that enables it to react to goal changes.

Finally, our work is also related to cognitive architectures and goal reasoning agents. Some cognitive architectures (Langley et al., 2009) use rules of the form ***if*** *conditions* ***then*** *goal*, which trigger the next goals to achieve depending on the current conditions. Here conditions can be broadly constructed to include annotations about the world state and actions in the current plan. The continuous-concept matching employed by Choi (2011) extends this representation to permit arbitration between current goals based on priority values dynamically computed from the degree of match offered by a goal's conditions. This serves a similar role to the fitness functions employed by M-ARTUE, but is based on domain-specific rules encoded in the agent's conceptual knowledge, whereas M-ARTUE employs domain-independent motivators. Others, such as MADBot (Coddington et al., 2005) or ARTUE (Klenk et al., 2013), represent motivations using domain knowledge to encode thresholds or conditions for known variables that the agent can observe. Thus, the goal selection knowledge is hard-coded in the rules. In contrast to these efforts, motivated agents prioritize goals according to the different motivators. This provides flexibility for learning because the relation between goals and goal selection is not hardcoded. Other goal reasoning systems such as LGDA (Jaidee et al., 2011) use reinforcement learning (RL) techniques to learn goal selection knowledge. Since these systems are guided by a user-defined reward function, they can be viewed as achieving social motivations in our parlance. As RL systems they

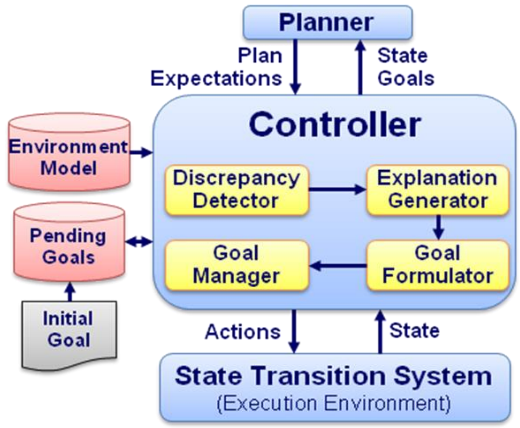


Figure 2. The Goal-Driven Autonomy Model implemented by M-ARTUE

perform exploration and exploitation of the search space, which in this case is goal selection knowledge.

3. The M-ARTUE Agent

M-ARTUE is an extension of the ARTUE GR agent. ARTUE implements Goal-Driven Autonomy (GDA), a GR model extending Nau's (2007) model of online planning. The GDA model is depicted in Figure 2.

ARTUE is composed of an online Controller, which interacts with a Planner and a State Transition System $\Sigma = (S, A, F, \gamma)$, with possible states S, actions available to the agent A, exogenous events that may be triggered by the environment F, and state transition function $\gamma: S \times (A \cup F) \rightarrow S$. The Planner receives as input the current state s_c, the current goal g_c, and a model of the State Transition System M_Σ; it produces as output a sequence of actions $\langle a_{c+1}, \ldots, a_{c+n} \rangle$ and a corresponding sequence of expectations $\langle x_{c+1}, \ldots, x_{c+n} \rangle$, where x_i is the state expected to follow a_i during execution.

As the agent executes a plan returned by the Planner, it performs four GR steps:

- **Notable Event Detection:** The agent executes the current plan and compares observed states from the environment with the sequence of expected states produced by the Planner. If the current observation differs meaningfully from the expected state, the agent notes the discrepancy and performs the following steps.
- **Notable Event Explanation:** The agent produces an explanation for a given discrepancy, where the explanation is frequently an adjustment to the agent's beliefs that incorporates unobserved (but abduced) facts and exogenous events.
- **Goal Formulation:** The agent produces a new goal, if necessary, that is deemed an appropriate response to a given explanation.
- **Goal Management:** The agent selects a goal or goals to pursue from goals that were formulated during a current or previous GR sequence.

M-ARTUE extends ARTUE by performing goal selection (i.e., goal formulation and goal management) through the application of motivators. When a notable event is detected and explained, M-ARTUE evaluates all possible goals for the agent using its motivators and selects the goal with the best combined score. Although the goal-driven autonomy model is planner agnostic, ARTUE implementations have historically used hierarchical task network (HTN) planners. Thus, possible goals for M-ARTUE are enumerated from a list of tasks in an HTN that are designated as top-level tasks for the domain. Any grounding of such a task in the current state for which a plan can be constructed is considered a possible goal, and M-ARTUE uses the constructed plans to evaluate their respective goals.

3.1 Motivators

Each motivator calculates an *urgency* value that indicates how important it is to fulfill its current needs. Urgency is defined as a function $u_m: S -> \mathbb{R}$, which expresses how urgent a particular motivator m's needs are in the current state $s_c \in S$. Each motivator evaluates the *fitness* of each goal g for satisfying its domain-independent needs by applying a motivator-specific fitness function $f_m: \langle x_c, x_{c+1}, \ldots x_{c+n} \rangle \rightarrow \mathbb{R}$ to the expectations $x_{c+1}, \ldots x_{c+n}$ generated by the Planner. Finally, for each goal, a weighted sum over the motivators is calculated, defined as:

$$fitness(g) = \sum_m u_m(s_c) \times f_m\big(Expectations(g, s_c)\big),$$

where g is a goal and $Expectations(g, s_c)$ is the list of expectations X returned by the Planner when given a goal g in state s_c. The goal g with the highest $fitness(g)$ is selected.

3.1.1 Social Motivator

The Social Motivator captures the need to achieve the user-designated goals. Currently, these are represented by a list of state conditions that must be true to satisfy a user-designated goal. The Social Motivator's urgency is a sawtooth function that increases over time until a user-defined goal is fulfilled. This function biases goal selection toward social conditions when they have not been achieved in some time. It is defined by the function:

$$u_{social}(s_c) = \begin{cases} C_{social} u_{social}(s_{c-1}), if\ R(s_c) \le R(s_{c-1}) \\ 0.1, if\ R(s_c) > R(s_{c-1}) \end{cases},$$

where s_c is the state at the time of goal selection, $R(s_c)$ is the percentage of user-provided goals that have been satisfied in s_c or some prior state $s_i (i < c)$ visited, and $C_{social} > 1$ is a constant of social motivation that is tuned to the domain.

The fitness function for the Social Motivator biases goal selection toward goals that achieve the most social conditions with the fewest actions. It is calculated as:

$$f_{social}(X) = C_{social-fitness} \frac{R(x_{c+n}) - R(s_c)}{n},$$

where X is the sequence of expected states as defined above, n is the plan's length, $C_{social-fitness}$ is a constant of social fitness that is tuned to the domain, and x_{c+n} is the expected state after the plan executes.

3.1.2 Exploration Motivator

The urgency of the Exploration Motivator is biased to increase when the most recent action has not visited a new unique state, and to be large when fewer states overall have been visited (i.e., exploration is most valued when little to no exploration has been done). It is defined as:

$$u_{exploration}(s_c) = 1 - \frac{V(s_0, s_1, \dots, s_c)}{V(s_0, s_1, \dots s_{c-1}) + C_{exploration}},$$

where $V(S)$ is the number of distinct states in a list S and $C_{Exploration}$ is a constant of exploration that is tuned to the domain.

The fitness function biases goal selection toward goals that visit the most new unique states per action. This function is defined as:

$$f_{exploration}(X) = \frac{V(s_0, s_1, \dots s_c, x_{c+1}, x_{c+2}, \dots x_{c+n}) - V(s_0, s_1, \dots s_c)}{n}.$$

3.1.3 Opportunity Motivator

The Opportunity Motivator tries to maximize the agent's opportunity to act throughout plan execution, thus helping the agent to prepare to fulfill future goals. This is evaluated in terms of two factors: (1) the branching factor in a given state and (2) the availability of resources relative to their historical averages. These factors are combined to determine this motivator's urgency, which biases selection toward opportunity-increasing goals when the agent cannot execute as many actions or it does not possess as many resources as have been available historically. This function is defined as:

$$u_{opportunity}(s_c) = \left[\left(1 - \frac{N(s_c)}{\max\limits_{0 \le i < c} N(s_i)}\right) + (1 - L(s_c))\right]/2,$$

where $N(s)$ is the number of available actions, and $L(s)$ is the level of resources relative to historical resource levels. A domain defines a set of k resources, each of which has a state-based level $v_r(s)$. Function $L(s)$ is defined in terms of these levels as $L(s_c) = (\sum_{r=1}^{k}[v_r(s_c)/a_r(s_c)])/k$, where $a_r(s_c) = \frac{\sum_{i=1}^{c-1} v_r(s_i)}{c-1}$ is the mean of all prior values for $v_r(s)$.

The Opportunity Motivator's fitness function biases goal selection toward goals that have the most actions available per expected state, and leaves the agent with the most resources and actions available when the goal is achieved. This function is defined as:

$$f_{opportunity}(X) = \frac{\left(\left[\sum_{j=0}^{n-1} N(x_{c+j})\right] + [w \times N(x_{c+n})]\right)}{(n+w)N(s_c)} + L(x_{c+n}) - L(s_c) - 1,$$

where $w \geq 1$.

4. Experiments and Discussion

We performed experiments to evaluate two hypotheses:

- **H0:** Varying motivator weights will affect agent performance.

- **H1:** Agent performance will vary with motivator weights in a predictable fashion as certain properties of the evaluation scenario change. Specifically, the scenario properties we investigated are:
    - *Initial observability:* When observability is low, we expect the number of goals achieved to be greater when the relative weight of the Exploration Motivator increases, encouraging the agent to observe more about its environment. Conversely, when observability is high we expect the number of goals achieved to be greater when the relative weight of the Social Motivator increases.
    - *Resource capacity:* When the agent has greater resource capacity, we expect the number of goals achieved to increase as the relative weight of the Opportunity Motivator increases, encouraging the agent to gather as many resources as possible. Conversely, when resource capacity is low, we expect the number of goals achieved to be greater when the relative weight of the Social Motivator increases.

To test these hypotheses, we ran M-ARTUE on scenarios from the Metric Transportation domain, a modified version of the Logistics Transportation domain. In this domain, an agent is given an initial set of goals by a user, who directs it to deliver a specified set of packages to a set of discrete destinations, which are located (as nodes) in a partially-connected graph. To achieve these goals, the agent uses trucks and airplanes to move packages. Our modified domain omits airports and airplanes, but includes a fuel function on trucks (i.e., a given truck's current fuel level), which decreases as the truck moves between connected locations according to a cost function defined on the connections. Additionally, our modifications permit the scenario author to initially hide some connections between locations, some gas stations, and some packages' locations. M-ARTUE discovers these hidden facts through the occurrence of observation events when a truck moves to a relevant location (i.e., one of the connected locations, a gas station's location, or a package's location, respectively). To guide M-ARTUE's goal selection and HTN planner, we created an HTN definition encompassing top-level tasks that allow the agent to deliver a package to a particular location, drive a truck to a particular location, refuel a truck, or do nothing.

We randomly generate scenarios in this domain according to parameters controlling: the size of the graph; the number of trucks, packages, and gas stations; the amount of fuel available to the trucks initially and after fueling at stations; and the connectedness of the graph. These parameters control the difficulty of the agent's planning problems. We also individually specify percentages of connections, packages, and gas stations that will be visible to the agent initially. These parameters impact the difficulty of the agent's goal-achievement problem. To evaluate the agent's performance, we use as a metric the fraction of total user goals achieved (i.e., the number of packages successfully delivered to their destinations).

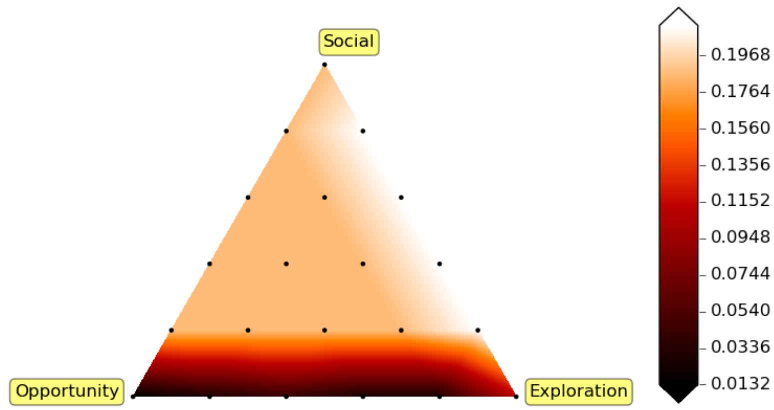


Figure 3. Average fraction of user goals achieved with high initial observability across 25 trials (maximum 500 actions) at indicated weight-mixture points

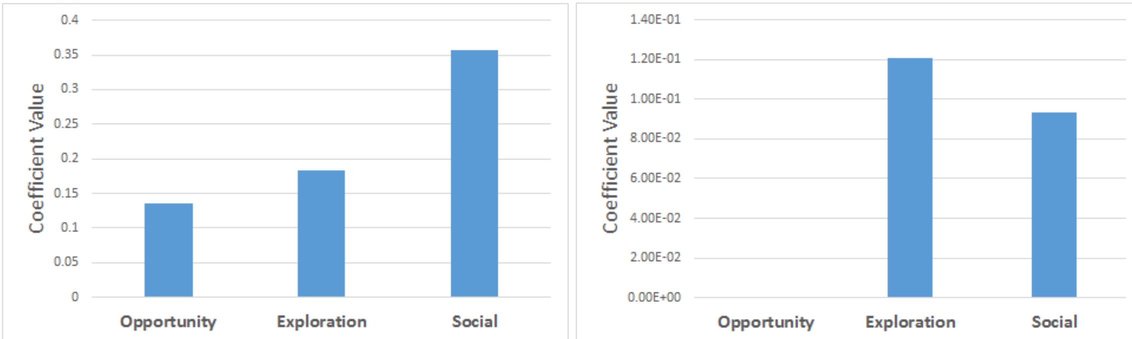
4.1 Motivator Weights and Agent Performance

To test hypothesis H0, we generated several scenarios with the same parameters. Specifically, for this test we adopted the scenario parameters $n_{locations} = 20$, $n_{trucks} = 3$, $n_{packages} = 3$, $n_{stations} = 3$, and $n_{connections} = 3$, with all packages and gas stations initially visible to the agent, and 70% of edges initially known to the agent. We generated 25 scenarios using these values. In each scenario, we evaluated the agent's performance at different weight mixtures corresponding to combinations that sum to 1 of the values (0, 0.2, 0.4, 0.6, 0.8, 1) for all motivators. We limited the test to 500 actions on each scenario-weight pair. (Since the agent can continue indefinitely, we chose 500 actions to provide a reasonable tradeoff of running time and goal achievement.) The average fraction of user goals accomplished at each weight mixture is shown in Figure 3.

Visually, the agent performs best in a region dominated by non-zero values of the Social Motivator. Additionally, the agent's performance is better when the Exploration Motivator is weighted more heavily than the Opportunity Motivator, indicating that exploration may improve agent performance by revealing even a few hidden environmental features (connections between locations, in this case). (Occasionally, the agent may deliver a package simply by trying different exploratory actions, even when the Social Motivator has no weight, as can be seen by the slight improvement in performance at the Exploration Motivator's corner.) An analysis of variance on the average agent performance indicates that the performance differs significantly across the range of weights ($p < 5 \times 10^{-8}$ for all components), supporting hypothesis H0.

4.2 Motivator Weights and Scenario Properties

To test hypothesis H1, we altered the 25 scenarios described in Section 4.1 to provide contrast in the desired scenario properties.



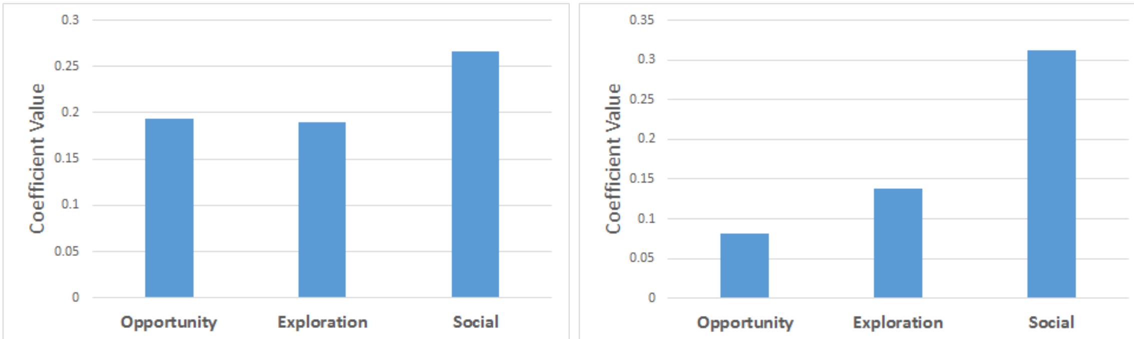
(a) Scenarios with high observability *(b)* Scenarios with low observability

Figure 4. Coefficients of motivator weight for agent performance

- **Observability:** We compared the original scenarios with modified versions that initially revealed none of the packages, gas stations, or connections, except those relevant to the trucks' starting locations.
- **Resource capacity:** We altered the scenarios to provide complete observability of the graph (to avoid any impact the Exploration Motivator might have on resource consumption by revealing shorter routes). We then compared these completely-observable scenarios using the original resource levels with modified versions that provide higher initial fuel levels and fuel capacity (95 and 150, compared to 25 and 40, respectively).

For each alteration to the scenarios, we evaluated the agent as in Section 4.1, using the same weight-mixture points and limiting the agent to 500 actions. We then compared the motivators' contributions to agent performance under the differing scenario properties by fitting a canonical linear mixture model, $y = \sum \beta_i x_i$, to the data for the original scenarios and the alterations. The results were as follows:

- **Observability:** Figure 4 depicts the motivator coefficients β_i in the linear fit model. These values indicate how strongly each motivator is correlated with agent performance in the 25 scenarios for the original high-observability scenarios and the matching low-observability scenarios. In the high-observability scenarios, the Social Motivator correlates most strongly with agent performance. By contrast, the Exploration Motivator correlates most strongly with agent performance in the low-observability scenarios, supporting our hypothesis that the importance of exploration increases (i.e., the agent achieves more goals when the Exploration Motivator is heavily weighted) when the environment exhibits low initial observability, as the agent cannot achieve user goals without discovering routes and gas stations, and the need to discover those features outweighs the need to conserve resources. In fact, in this extreme scenario, conserving resources contributed nothing to agent performance. The smaller numerical values of the coefficients in the low-observability scenarios are due to lower overall agent performance, as the environment is more challenging. (Note that, while the linear fit was significant for all three components in the high observability scenarios, it was not significant for the Opportunity Motivator in the low observability scenarios, supporting the conclusion that the Opportunity Motivator was not a significant contributor to agent performance in those experiments.)



(a) Scenarios with high resource capacity

(b) Scenarios with low resource capacity

Figure 5. Coefficients of motivator weight for agent performance

- **Resource capacity:** Figure 5 depicts the motivator coefficients β_i in the linear fit model. These values indicate motivator correlation with agent performance in the 25 high-resource scenarios and the matching low-resource scenarios. While the Social Motivator is the dominant contributing motivator to agent performance in all scenarios, the Opportunity Motivator contributes much more to agent performance in the high-capacity scenarios than in the low-capacity scenarios, supporting our hypothesis that the importance of resource acquisition increases as a function of the agent's ability to gather and retain more resources to assist in achieving goals. (The linear fit was significant for all components in these experiments.)

5. Conclusion and Future Work

The paradox of Buridan's ass exemplifies the need to balance between an agent's motivations. We reviewed a GR agent (M-ARTUE) that reasons with Social, Exploration, and Opportunity motivators to deliberate among goal choices. We introduced the Metric Transportation Domain to test the motivators' value in scenarios with varying properties, and showed that (1) varying the motivators' relative weights can impact agent performance and (2) the relative importance of each motivator is context-dependent.

In future work we will investigate the impact of motivator weight settings in other domains (e.g., an underwater vehicle domain and a Mars rover domain). We will also investigate the effects of the motivator weight settings in other extreme scenarios (e.g., when resource consumption is extremely low or resource availability is extremely high). We will use the results of these investigations to identify further predictable domain and problem characteristics that affect motivators' correlation with agent performance, and we will pursue the creation of a more formal model of domain and problem characteristics and their interaction with motivator weights. We will also investigate the effect of motivator weights using alternative metrics of agent performance (e.g., how quickly user goals are achieved and how many resources the agent expends while achieving them). We will investigate the use of other motivators (e.g., a directed information motivator). Finally, we will investigate how an agent can learn weight settings that will enable it to outperform its behavior using fixed equal weights. We will do so using fixed environmental conditions, as we used for these experiments (e.g., resource capacity is static throughout an experiment), and environments in which these conditions may change dynamically.

Acknowledgements

This work was supported in part by the National Science Foundation.

References

Coddington, A.M., Fox, M., Gough, J., Long., D., & Serina, I. (2005). MADbot: A motivated and goal directed robot. *Proceedings of the Twentieth National Conference on Artificial Intelligence* (pp. 1680-1681). Pittsburgh, PA: AAAI Press.

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

Do, M.B., Benton, J., van den Briel, M., & Kambhampati, S. (2007). Planning with goal utility dependencies. *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 1872-1878).

Draper, D., Hanks, S., & Weld, D. S. (1994). Probabilistic planning with information gathering and contingent execution. *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems* (pp. 31-36). Chicago, IL: AAAI Press.

Ghallab, M., Nau, D.S., & Traverso, P. (2004). *Automated planning: Theory and practice*. San Mateo, CA: Morgan Kaufmann.

Jaidee, U., Muñoz-Avila, H., & Aha, D.W. (2011). Integrated learning for goal-driven autonomy. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. Barcelona, Spain.

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.

Kuffner, J.J. (2004). Efficient optimal search of uniform-cost grids and lattices. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1946-1951). Sendai, Japan: IEEE Press.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*(2), 141-160.

Muñoz-Avila, H., Aha, D.W., Jaidee, U., Klenk, M., & Molineaux, M. (2010). Applying goal directed autonomy to a team shooter game. *Proceedings of the Twenty-Third Florida Artificial Intelligence Research Society Conference* (pp. 465-470). Daytona Beach, FL: AAAI Press.

Nau, D.S. (2007). Current trends in automated planning. *AI Magazine*, *28*(4), 43–58.

Pryor, L., & Collins, G. (1996). Planning for contingencies: A decision-based approach. *Journal of Artificial Intelligence Research*, *4*, 287-339.

Roberts, M., Vattam, S., Alford, R., Auslander, B., Karneeb, J., Molineaux, M., Apker, T., Wilson, M., McMahon, J., & Aha, D.W. (2014). Iterative goal refinement for robotics. In A. Finzi & A. Orlandini (Eds.) *Planning and Robotics: Papers from the ICAPS Workshop*. Portsmouth, NH: 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, BC, Canada: AAAI Press.

Stentz, A. (1995). The focussed D* algorithm for real-time replanning. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1652-1659). Montreal, Quebec, Canada: Morgan Kaufmann.

Vattam, S., Klenk, M., Molineaux, M., & Aha, D.W. (2013). Breadth of approaches to goal reasoning: A research survey. In D.W. Aha, M.T. Cox, & H. Muñoz-Avila (Eds.) *Goal Reasoning: Papers from the ACS Workshop* (Technical Report CS-TR-5029). College Park, MD: University of Maryland, Department of Computer Science.

Veloso, M.M. (1994). *Planning and learning by analogical reasoning*. Springer: Berlin.

Williamson, M., & Hanks, S. (1994). Optimal planning with a goal-directed utility model. *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems* (pp. 176-181). Chicago, IL: AAAI Press.