Metareasoning for Safe Decision Making in Autonomous Systems

Justin Svegliato[1], Connor Basich[2], Sandhya Saisubramanian[3], and Shlomo Zilberstein[2]

***Abstract*— Although experts carefully specify the high-level decision-making models in autonomous systems, it is infeasible to guarantee safety across every scenario during operation. We therefore propose a *safety metareasoning system* that optimizes the *severity* of the system's safety concerns and the *interference* to the system's task: the system executes in parallel a *task process* that completes a specified task and *safety processes* that each address a specified safety concern with a *conflict resolver* for arbitration. This paper offers a formal definition of a safety metareasoning system, a recommendation algorithm for a safety process, an arbitration algorithm for a conflict resolver, an application of our approach to planetary rover exploration, and a demonstration that our approach is effective in simulation.**

I. Introduction

While planning and robotics experts carefully design, build, and test the models used by autonomous systems for high-level decision making, it is infeasible for these models to ensure safety across every scenario within the domain of operation [1]. This is due to the challenge of specifying comprehensive decision-making models given the complexity of the state space or action space, a lack of information about the environment, or a misunderstanding of the limitations of the autonomous system [2]. For example, a courier robot could use a decision-making model with features for safely interacting with different types of doors but not for navigating a crosswalk, which increases the risk of endangering people, damaging property, or breaking the courier robot [3]. Therefore, as autonomous systems grow in independence and sophistication [4], it is critical to give them the ability to maintain and restore safety during operation.

A naive approach to giving an autonomous system the ability to maintain and restore safety is to use a comprehensive decision-making model with every feature needed to cover every scenario within the domain of operation. This model, however, would suffer from two main drawbacks in real world environments [1]. First, the model would simply be infeasible to design due to the intractability of complex environments. Second, even if it were feasible to design, the model would likely be infeasible to solve with exact or even approximate methods due to the urgency of real-time environments. Hence, to avoid the infeasibility of a monolithic model, this paper offers a scalable framework for safe decision making in autonomous systems that decouples the system into a primary process with features required to achieve its main goal and secondary processes each with features required to respond to a particular hazard.

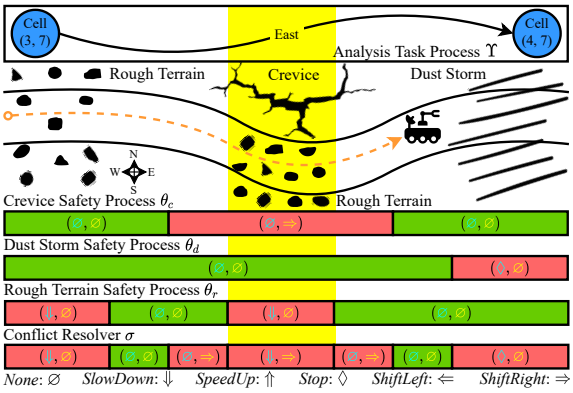

Fig. 1. A planetary rover executes a task process Υ that analyzes different points of interest within a region of a planet and safety processes θ_c, θ_d, and θ_r that address crevices, dust storms, and rough terrain with a conflict resolver σ for arbitration. Consider the highlighted time slice that shows the planetary rover completing the analysis task while addressing crevices, dust storms, and rough terrain. Intuitively, (1) the task process performs the *East* action starting in the cell $(3, 7)$ and ending in the cell $(4, 7)$, (2) the safety processes θ_c, θ_d, and θ_r recommend the parameters $(\varnothing, \Rightarrow)$, $(\varnothing, \varnothing)$, and $(\Downarrow, \varnothing)$, that can adjust the wheel rotation rate and the steering of the *East* action being performed by the task process Υ, and (3) the conflict resolver σ selects the optimal parameter $(\Downarrow, \Rightarrow)$ that adjusts the *East* action being performed by the task process Υ given the parameters $(\varnothing, \Rightarrow)$, $(\varnothing, \varnothing)$, and $(\Downarrow, \varnothing)$ recommended by the safety processes θ_c, θ_d, and θ_r.

Several areas of work that focus on safety in autonomous systems have seen recent attention [4]. First, methods avoid *negative side effects* that cause a system to interfere with its environment (e.g., by adding an extra term to its objective function [5], [6] or modifying its decision-making model based on human feedback [7]). Next, methods mitigate *reward hacking* that cause a system to game its reward function (e.g., by applying ethical constraints to its behavior [8], [9], [10], [11], [12], [13] or treating its reward function as an observation of its true objective function [14], [15], [16]). Finally, methods handle *distributional change* that cause a system to perform poorly in a new environment that differs from its original environment (e.g., by detecting anomalies using Monte Carlo methods based on particle filters [17], [18], [19] or multiple model estimation based on neural networks [20], [21]). However, while these areas are critical to safety, this paper focuses on tweaking the operation of an autonomous system for safe decision making.

We propose a disciplined, decision-theoretic metareasoning approach to safe decision making in autonomous systems. A *safety metareasoning system* executes in parallel a *task process* that completes a specified task and *safety processes* that each address a specified safety concern with a *conflict resolver* for arbitration. Like a standard autonomous system, the task process completes a specified task by performing an action in its current state following its policy.

This work was supported in part by the NSF GRFP (DGE-1451512) and the NSF (IIS-1954782 and IIS-1813490).

[1]University of California, Berkeley, CA, USA. Email: jsvegliato@berkeley.edu

[2]University of Massachusetts, Amherst, MA, USA. Email: {cbasich,shlomo}@cs.umass.edu

[3]Oregon State University, Corvallis, OR, USA. Email: sandhya.sai@oregonstate.edu

However, at fixed intervals as the task process performs each action, there are two operations that are not considered by a standard autonomous system. First, the safety processes each address a specified safety concern by recommending a rating over a set of parameters in its current state that can adjust the action being performed by the task process. Second, the conflict resolver for arbitration selects the optimal parameter that will adjust the action being performed by the task process given the ratings over the set of parameters recommended by the safety processes. Our experiments on the planetary rover exploration domain illustrated in Figure 1 highlights that our approach optimizes the *severity* of safety concerns (the danger of particular hazards) and the *interference* to the task (the overhead of safety on the main goal).

Our main contributions are: (1) a formal definition of a safety metareasoning system, (2) a recommendation algorithm for a safety process, (3) an arbitration algorithm for a conflict resolver, (4) an application of our approach to planetary rover exploration, and (5) a demonstration that our approach is effective in simulation.

II. Background

A *Markov decision process* (MDP) is a decision process for reasoning in fully observable, stochastic environments [22]. An MDP is described by a tuple $\langle S, A, T, R \rangle$. The set of states is S. The set of actions is A. The transition function $T : S \times A \times S \to [0, 1]$ represents the probability of reaching a state $s' \in S$ after performing an action $a \in A$ in a state $s \in S$. The reward function $R : S \times A \to \mathbb{R}$ represents the expected immediate reward of performing an action $a \in A$ in a state $s \in S$. A solution to an MDP is a policy $\pi : S \to A$ indicating that an action $\pi(s) \in A$ should be performed in a state $s \in S$. A value function $V^\pi : S \to \mathbb{R}$ represents the expected discounted cumulative reward $V^\pi(s) \in \mathbb{R}$ of starting in a state $s \in S$ following a policy $\pi : S \to A$ for a given discount factor $0 \leq \gamma < 1$. An optimal policy $\pi^* : S \to A$ maximizes the expected discounted cumulative reward $V^*(s) \in \mathbb{R}$ for each state $s \in S$ corresponding to an optimal value function $V^*(s) = \max_{a \in A} \left[R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \right]$.

Value iteration is a common algorithm that computes the optimal value function V^* [23]. It begins with an optimal 0-horizon value function V_0^*. It then builds an optimal $(t + 1)$-horizon value function V_{t+1}^* from an optimal t-horizon value function V_t^* by using the Bellman backup operator, $V_{t+1}^* = \max_{a \in A} \left[R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_t^*(s') \right]$, for each time step t until the condition $\|V_{t+1} - V_t\|_\infty < \epsilon \frac{(1-\gamma)}{\gamma}$ is satisfied for a given convergence threshold ϵ.

III. Metareasoning for Safety

In this section, we introduce a *safety metareasoning system* that executes in parallel a *task process* that completes a specified task and *safety processes* that each address a specified safety concern with a *conflict resolver* for arbitration. Note that this is a typical form of metareasoning [24], [25], [26], [27], [28] in that meta-level processes (*safety processes*) monitor and control an object-level process (*task process*).

1) Task Processes: The task process completes a specified task by performing an action in its current state following its policy. The representation of the task process must reflect the properties of the task. This paper represents a task process as an MDP, a decision process for tasks with full observability, because it is a standard model in planning and robotics [29]. However, it is possible to use different classes of decision processes for tasks with partial observability [30] or start and goal states [31]. We define the task process below.

Definition 1. *The **task process**, represented by an MDP $\Upsilon = \langle S, A, T, R \rangle$, performs an action $a = \pi(s) \in A$ in a state $s \in S$ following a policy π to complete a specified task.*

Example. To complete the analysis task, the planetary rover in the highlighted time slice of Figure 1 executes the task process Υ that performs the *East* action starting in the cell $(3, 7)$ and ending in the cell $(4, 7)$.

2) Safety Processes: A safety process addresses a specified safety concern by recommending a rating over a set of parameters in its current state that can adjust the action being performed by the task process. The representation of a safety process is a variant of an MDP with several attributes: a set of states that describe the safety concern, a set of parameters that can adjust the action being performed by the task process, a transition function that reflects the dynamics of the world, a severity function that reflects the severity of the safety concern, and an interference function that reflects the interference to the task. We define a safety process below.

Definition 2. *A **safety process**, represented by a variant of an MDP $\theta = \langle \bar{S}, \bar{P}, \bar{T}, \phi, \psi \rangle \in \Theta$, recommends a rating $\rho_{\bar{s}}^\theta$ over a set of parameters $\bar{P}$ in a state $\bar{s} \in \bar{S}$ that can adjust the action $a \in A$ being performed by the task process Υ to address a specified safety concern.*

- *$\bar{S}$ is a set of states that describe the safety concern.*
- *$\bar{P} = \bar{P}_1 \times \bar{P}_2 \times \cdots \times \bar{P}_N$ is a set of parameters such that each parameter factor $\bar{P}_i$ adjusts the action $a \in A$ being performed by the task process Υ with a $\varnothing \in \bar{P}_i$ symbol that indicates no adjustment.*
- *$\bar{T} : \bar{S} \times \bar{P} \times \bar{S} \to [0, 1]$ is a transition function that represents the probability of reaching a state $\bar{s}' \in \bar{S}$ after using a parameter $\bar{p} \in \bar{P}$ in a state $\bar{s} \in \bar{S}$.*
- *$\phi : \bar{S} \to \{1, 2, \ldots, L\}$ is a severity function that represents the severity of the safety concern in a state $\bar{s} \in \bar{S}$ such that 1 is the lowest level and L is the highest level where a severity level $1 \leq \ell \leq L$ is strictly preferred to a severity level $1 \leq \ell + 1 \leq L$.*
- *$\psi : \bar{P} \to \mathbb{R}^+$ is an interference function that represents the interference of a parameter $\bar{p} \in \bar{P}$ on the action $a \in A$ being performed by the task process Υ.*

Example. To address crevices, dust storms, and rough terrain, the planetary rover in the highlighted time slice of Figure 1 executes the safety processes θ_c, θ_d, and θ_r that recommend the parameters $(\varnothing, \Rightarrow)$, $(\varnothing, \varnothing)$, and $(\Downarrow, \varnothing)$ that can adjust the wheel rotation rate and the steering of the *East* action being performed by the task process Υ.

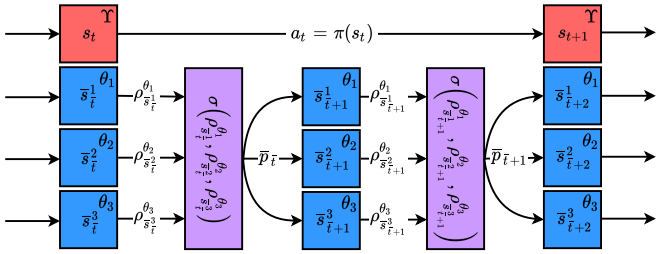

Fig. 2. A safety metareasoning system that has the task process in *red*, the safety processes in *blue*, and the conflict resolver in *purple*.

Each safety process recommends *a rating over a set of parameters* instead of only *a parameter*. For a given state, this rating contains $|L|+1$ values for each of the $|\bar{P}|$ parameters: the *expected discounted frequency of each severity level* and the *expected discounted cumulative interference* that would be incurred by the safety process if it were to use that parameter in that state. We define this rating below.

Definition 3. *A **rating**, $\rho^{\theta}_{\bar{s}}$, over a set of parameters $\bar{P}$ in a state $\bar{s} \in \bar{S}$ recommended by a safety process $\theta \in \Theta$ is represented by the following $|\bar{P}| \times (|L|+1)$ matrix:*

$$\rho^{\theta}_{\bar{s}} = \begin{bmatrix} \Phi^{\theta}_{\bar{s},\bar{p}_1}[1] & \Phi^{\theta}_{\bar{s},\bar{p}_1}[2] & \dots & \Phi^{\theta}_{\bar{s},\bar{p}_1}[L] & \Psi^{\theta}_{\bar{s},\bar{p}_1} \\ \Phi^{\theta}_{\bar{s},\bar{p}_2}[1] & \Phi^{\theta}_{\bar{s},\bar{p}_2}[2] & \dots & \Phi^{\theta}_{\bar{s},\bar{p}_2}[L] & \Psi^{\theta}_{\bar{s},\bar{p}_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi^{\theta}_{\bar{s},\bar{p}_N}[1] & \Phi^{\theta}_{\bar{s},\bar{p}_N}[2] & \dots & \Phi^{\theta}_{\bar{s},\bar{p}_N}[L] & \Psi^{\theta}_{\bar{s},\bar{p}_N} \end{bmatrix}.$$

*When a safety process $\theta \in \Theta$ uses a parameter $\bar{p} \in \bar{P}$ in a state $\bar{s} \in \bar{S}$, the **expected discounted frequency of each severity level** $1 \leq \ell \leq L$ incurred is $\Phi^{\theta}_{\bar{s},\bar{p}}[\ell] = [\phi(\bar{s}) = \ell] + \gamma \sum_{\bar{s}' \in \bar{S}} \bar{T}(\bar{s},\bar{p},\bar{s}') \min_{\bar{p}' \in \bar{P}} \Phi^{\theta}_{\bar{s}',\bar{p}'}[\ell]$ and the **expected discounted cumulative interference** incurred is $\Psi^{\theta}_{\bar{s},\bar{p}} = \psi(\bar{p}) + \gamma \sum_{\bar{s}' \in \bar{S}} \bar{T}(\bar{s},\bar{p},\bar{s}') \min_{\bar{p}' \in \bar{P}} \Psi^{\theta}_{\bar{s}',\bar{p}'}$. Note that the operator $[\cdot]$ denotes Iverson bracket notation.*

3) Conflict Resolvers: The conflict resolver for arbitration selects the optimal parameter that will adjust the action being performed by the task process given the ratings over the set of parameters recommended by the safety processes. Intuitively, if no safety process or only one safety process encounters its safety concern, there is no need for arbitration. However, if multiple safety processes encounter their safety concerns, the conflict resolver arbitrates by selecting the parameter that optimally addresses each safety concern. The representation of the conflict resolver is a function that maps the ratings over the set of parameters recommended by the safety processes to a parameter. We define the conflict resolver below.

Definition 4. *The **conflict resolver**, $\sigma : \rho^{\theta_1}_{\bar{s}^1} \times \rho^{\theta_2}_{\bar{s}^2} \times \cdots \times \rho^{\theta_n}_{\bar{s}^n} \to \bar{P}$, selects the optimal parameter $\bar{p} \in \bar{P}$ that adjusts the action $a \in A$ being performed by the task process Υ given the ratings $\rho^{\theta_i}_{\bar{s}^i}$ over the set of parameters $\bar{P}$ recommended by the safety processes $\theta_i \in \Theta$ for arbitration.*

Example. For arbitration, the planetary rover in the highlighted time slice of Figure 1 uses the conflict resolver σ that selects the optimal parameter $(\Downarrow, \Rightarrow)$ that adjusts the *East* action being performed by the task process Υ given the parameters $(\varnothing, \Rightarrow)$, $(\varnothing, \varnothing)$, $(\Downarrow, \varnothing)$ recommended by the safety processes θ_c, θ_d, and θ_r.

The optimal parameter selected by the conflict resolver satisfies a lexicographic objective function. First, in the order of decreasing severity level, this parameter must *minimize* the *maximum* expected discounted frequency of each severity level incurred across all safety processes (minimize the maximum anticipated danger of particular hazards). Second, this parameter must *minimize* the *maximum* expected discounted cumulative interference incurred across all safety processes (minimize the maximum anticipated overhead of safety on the main goal). Formally, given each rating $\rho^{\theta_i}_{\bar{s}^i}$ over the set of parameters $\bar{P}$ recommended by each safety process $\theta_i \in \Theta$ in its state $\bar{s}^i \in \bar{S}^i$, we define this function below.

$$\min_{\bar{p} \in \bar{P}} \Big[\max_{\theta_i \in \Theta} \big[\Phi^{\theta_i}_{\bar{s}^i,\bar{p}}[L] \succ \Phi^{\theta_i}_{\bar{s}^i,\bar{p}}[L-1] \succ \cdots \succ \Phi^{\theta_i}_{\bar{s}^i,\bar{p}}[1] \succ \Psi^{\theta_i}_{\bar{s}^i,\bar{p}} \big] \Big]$$

4) Safety Metareasoning Systems: We provide a complete description of a safety metareasoning system below.

Definition 5. *A **safety metareasoning system**, $\langle \Upsilon, \Theta, \sigma \rangle$, runs in parallel a task process Υ that completes a specified task and safety processes Θ that each address a specified safety concern with a conflict resolver σ for arbitration.*

Figure 2 summarizes a safety metareasoning system. There is a task transition for the task process Υ from the state $s_t \in S$ at time step $t \in H$ to the successor state $s_{t+1} \in S$ at time step $(t+1) \in H$ given the action $a_t = \pi(s_t) \in A$. During this task transition, there are many safety transitions for each safety process $\theta_i \in \Theta$ from the state $\bar{s}^i_{\bar{t}} \in \bar{S}^i$ at time step $\bar{t} \in \bar{H}$ to the successor state $\bar{s}^i_{\bar{t}'} \in \bar{S}^i$ at time step $\bar{t}' \in \bar{H}$. In each safety transition, each safety process $\theta_i \in \Theta$ recommends a rating $\rho^{\theta_i}_{\bar{s}^i_{\bar{t}}}$ over the set of parameter $\bar{P}$ to the conflict resolver σ. The conflict resolver σ then selects the optimal parameter $\bar{p}_{\bar{t}} \in \bar{P}$ that satisfies the lexicographic objective function. Once the optimal parameter $\bar{p}_{\bar{t}} \in \bar{P}$ is selected by the conflict resolver σ, the action $a_t = \pi(s_t) \in A$ of the task process Υ is adjusted in a way that reflects that optimal parameter. Notice that the task process Υ operates on course-grained time steps $t \in H$ while each safety process $\theta_i \in \Theta$ operates on fine-grained time steps $\bar{t} \in \bar{H}$ as the actions performed by the task process can continually be adjusted by the parameters recommended by the safety processes.

The actions of the task process and the parameters of the safety processes are tightly integrated. In particular, a safety metareasoning system must send an action and a parameter to a motion planner that computes motor commands that reflect performing the action subject to the constraints imposed by the parameter. Suppose that a planetary rover performs the *North* action with the $(\Downarrow, \Leftarrow)$ parameter for slowing down and shifting left. Here, the planetary rover must send the *North* action and the $(\Downarrow, \Leftarrow)$ parameter to the motion planner that must compute motor commands that move the planetary rover north subject to the constraints of slowing down and swerving left. Formally, it is possible to view an action $a \in A$ of a task process as a parameterized action $a[\bar{p}] \in A$ given a parameter $\bar{p} \in \bar{P}$ of the safety processes.

We now describe the two necessary algorithms of a safety metareasoning system in the following subsections.

Algorithm 1: The recommendation algorithm.

Input: The safety process $\theta = \langle \bar{S}, \bar{P}, \bar{T}, \phi, \psi \rangle$
Output: The matrix ρ^θ that is used to construct the rating $\rho^\theta_{\bar{s}}$ for each state $\bar{s} \in \bar{S}$ of the safety process θ

1 **for** $\ell \to L, L-1, \ldots, 1$ **do**
2 $\quad \Phi^\theta[\ell] \leftarrow 0_{\bar{S} \times \bar{P}}$
3 $\Psi^\theta \leftarrow 0_{\bar{S} \times \bar{P}}$
4 $\Lambda \leftarrow \emptyset$
5 **for** $\ell \to L, L-1, \ldots, 1$ **do**
6 $\quad \kappa(\bar{s}) \coloneqq [\phi(\bar{s}) = \ell]$
7 $\quad \Phi^\theta[\ell] \leftarrow \text{MODIFIEDVALUEITERATION}(\theta, \kappa, \Lambda)$
8 $\quad$ **for** $\bar{s}$ **in** $\bar{S}$ **do**
9 $\quad\quad \alpha \leftarrow \min_{\bar{p} \in \bar{P}} \Phi^\theta_{\bar{s},\bar{p}}[\ell]$
10 $\quad\quad$ **for** $\bar{p}$ **in** $\bar{P}$ **do**
11 $\quad\quad\quad$ **if** $\Phi^\theta_{\bar{s},\bar{p}}[\ell] > \alpha$ **then**
12 $\quad\quad\quad\quad \Lambda \leftarrow \Lambda \cup (\bar{s}, \bar{p})$
13 $\kappa(\bar{p}) \coloneqq \psi(\bar{p})$
14 $\Psi^\theta \leftarrow \text{MODIFIEDVALUEITERATION}(\theta, \kappa, \Lambda)$
15 **return** $\rho^\theta = \left[\Phi^\theta[1], \Phi^\theta[2], \ldots, \Phi^\theta[L], \Psi^\theta\right]$

A. Recommendation Algorithm

The *recommendation algorithm* in Algorithm 1 generates a matrix that is used to construct the rating for each state of a safety process (between the *blue* and *purple* objects in Figure 2). Given a safety process, this involves generating the *expected discounted frequency of each severity level* and the *expected discounted cumulative interference* incurred when using a parameter in a state for every state and parameter of the safety process. Note that this algorithm is run *offline* for each safety process before the operation of the system.

Initially, for each severity level and the interference, an $|\bar{S}| \times |\bar{P}|$ matrix is initialized (Lines 1-3). The $|\bar{S}| \times |\bar{P}|$ matrix for each severity level (Lines 5-7) and the interference (Lines 13-14) is then filled with its corresponding expected discounted values using *modified* value iteration that minimizes over *states* and *parameters* given a *cost* function instead of maximizing over *states* and *actions* given a *reward* function. Observe that the cost function $\kappa(\bar{s})$ is used to compute the expected discounted frequency of each severity level (Line 6) while the cost function $\kappa(\bar{p})$ is used to compute the expected discounted cumulative interference (Line 13). Finally, the $|\bar{S}| \times |\bar{P}|$ matrix for each severity level and the interference is stacked into an $|\bar{S}| \times |\bar{P}| \times (L+1)$ matrix (Line 15) that is used to construct the rating for each state of a safety process.

Most importantly, in order to satisfy the lexicographic objective function, a set of violating state-parameter pairs is initialized (Line 4). For each severity level, a state-parameter pair is added to the set of violating state-parameter pairs *if* that parameter in that state is worse than the optimal parameter in that state (Lines 8-12). The set of violating state-parameter pairs enables modified value iteration to forbid every state-parameter pair that did not satisfy the lexicographic objective function from the previous executions of modified value iteration (Lines 7 and 14).

We show the correctness and the worst-case time complexity of the recommendation algorithm below.

Algorithm 2: The arbitration algorithm.

Input: The ratings $\rho^{\theta_i}_{\bar{s}^i}$ in the current state $\bar{s}^i \in \bar{S}^i$ of the safety processes $\theta_i \in \Theta$
Output: A random optimal parameter $\bar{p} \in \bar{P}$

1 $\bar{P}^* \leftarrow \bar{P}$
2 **for** $\nu \to \Phi[L], \Phi[L-1], \ldots, \Phi[1], \Psi$ **do**
3 $\quad \alpha \leftarrow \min_{\bar{p} \in \bar{P}} \left[\max_{\theta_i \in \Theta} \nu^{\theta_i}_{\bar{s}^i,\bar{p}}\right]$
4 $\quad$ **for** $\bar{p}$ **in** $\bar{P}^*$ **do**
5 $\quad\quad \beta \leftarrow \max_{\theta_i \in \Theta} \nu^{\theta_i}_{\bar{s}^i,\bar{p}}$
6 $\quad\quad$ **if** $\beta > \alpha$ **then**
7 $\quad\quad\quad \bar{P}^* \leftarrow \bar{P}^* \setminus \{\bar{p}\}$
8 **return** $\text{RANDOM}(\bar{P}^*)$

Proposition 1 (Correctness)**.** *Algorithm 1 generates a matrix ρ^θ of the expected discounted frequency $\Phi^\theta_{\bar{s},\bar{p}}[\ell]$ of each severity level $1 \leq \ell \leq L$ and the expected discounted cumulative interference $\Psi^\theta_{\bar{s},\bar{p}}$ for each state $\bar{s} \in \bar{S}$ and parameter $\bar{p} \in \bar{P}$ of a safety process $\theta \in \Theta$ that satisfies the lexicographic objective function.*

Proof Sketch. Observe that there is an execution of a form of value iteration for each severity level and the interference in the order of the lexicographic objective function. It is known that standard value iteration without any set of violating state-parameter pairs would compute the corresponding expected discounted values for each severity level and the interference but may not satisfy the lexicographic objective function. However, by forbidding the set of violating state-parameter pairs, modified value iteration satisfies the lexicographic objective function. Thus, Algorithm 1 is correct. □

Proposition 2 (Time Complexity)**.** *Algorithm 1 has a worst-case time complexity of $O((L+1)|\bar{S}|^2|\bar{P}|)$.*

Proof Sketch. There are $L+1$ executions of value iteration that each have a time complexity of $O(|\bar{S}|^2|\bar{P}|)$ for a total time complexity of $O((L+1)|\bar{S}|^2|\bar{P}|)$. □

B. Arbitration Algorithm

The *arbitration algorithm* in Algorithm 2 implements the conflict resolver that selects the parameter that optimally addresses each safety process (between the *purple* and *blue* objects in Figure 2). Initially, a set of potentially optimal parameters is initialized (Line 1). Each severity level in the order of decreasing severity level followed by the interference is then processed (Line 2). To optimize the lexicographic objective function, the set of potentially optimal parameters is then pruned (Line 3-7). This involves computing the value of the parameter that minimizes the maximum respective expected discounted value over all safety processes (Line 3). With the value of this parameter, each parameter that has a maximum respective expected discounted value greater than that value is pruned (Line 4-7). Finally, a random optimal parameter that optimally addresses each safety process is selected (Line 8). Note that this algorithm is performed *online* during the operation of the system.

We show the correctness and the worst-case time complexity of the arbitration algorithm below.

Proposition 3 (Correctness)**.** *Algorithm 2 selects a random optimal parameter $\bar{p} \in \bar{P}$ that optimizes the lexicographic objective function given the ratings $\rho^{\theta_i}_{\bar{s}^i}$ in the current state $\bar{s}^i \in \bar{S}^i$ of the safety processes $\theta_i \in \Theta$.*

Proof Sketch. In the order of the lexicographic objective function, any parameter with a maximum expected discounted frequency greater than the optimal parameter for each severity level is pruned and any parameter with a maximum discounted cumulative interference greater than the optimal parameter for the interference is pruned. As this optimizes the lexicographic objective function, any remaining parameter is optimal. Hence, Algorithm 2 is correct. □

Proposition 4 (Time Complexity)**.** *Algorithm 2 has a worst-case time complexity of $O((L+1)|\bar{P}||\Theta|)$.*

Proof Sketch. There are L severity level pruning steps that each have a time complexity of $O(|\bar{P}||\Theta|)$ and an interference pruning step that has a time complexity of $O(|\bar{P}||\Theta|)$ for a total time complexity of $O((L+1)|\bar{P}||\Theta|)$. □

IV. Planetary Rover Exploration

We turn to an application of our approach to planetary rover exploration [32]: a planetary rover must analyze different points of interest within a region of a planet while addressing crevices, dust storms, and rough terrain.

The planetary rover has 4 internal components: a battery of a battery level $b \in B = \{0, 1, \ldots, M\}$ where 0 is a discharged battery and M is a charged battery, a rock analyzer of a health status $h_1 \in H_1 = \{\text{NOMINAL}, \text{ERROR}\}$, a soil analyzer of a health status $h_2 \in H_2 = \{\text{NOMINAL}, \text{ERROR}\}$, and an objective report $o \in O = \{\text{TRUE}, \text{FALSE}\}^I$ with an analysis status TRUE or FALSE for all points of interest I.

The planetary rover is within a region of a planet as an m by n grid where each cell is at a horizontal location $x \in X = \{1, 2, \ldots, n\}$ and a vertical location $y \in Y = \{1, 2, \ldots, m\}$ with weather of a type $w \in W = \{\text{LIGHT}, \text{DARK}\}$.

The planetary rover can perform 4 movement actions in each cell (x, y): it can move *north* to a cell $(x, y+1)$, *east* to a cell $(x+1, y)$, *south* to a cell $(x, y-1)$, or *west* to a cell $(x-1, y)$ if the new horizontal position is between 1 and n and the new vertical position is between 1 and m.

The planetary rover can perform 4 static actions in each cell (x, y): it can *reboot* its analyzers to set the health statuses of the rock analyzer h_1 and the soil analyzer h_2 to NOMINAL, *charge* its battery to the battery level $b' = (b+2)$ if the cell (x, y) has weather of a type $w = \text{LIGHT}$, *analyze* the cell (x, y) if the health statuses of the rock analyzer h_1 and the soil analyzer h_2 are set to NOMINAL, and *transmit* its data to complete the mission if the objective report is $o = \{\text{TRUE}\}^I$ with an analysis status TRUE for all points of interest I.

All actions discharge the battery to a battery level $b' = (b-1)$ and requires the battery to be at a battery level $b > 0$.

A. Task Process

We consider the task process, $\Upsilon = \langle S, A, T, R \rangle$, designed to complete the analysis task of the planetary rover. The set of states $S = X \times Y \times B \times H_1 \times H_2 \times O$ crosses a set of horizontal positions X, a set of vertical positions Y, a set of battery levels B, a set of rock analyzer health statuses H_1, a set of soil analyzer health statuses H_2, and a set of objective reports O. The set of actions $A = \{\uparrow, \rightarrow, \downarrow, \leftarrow, \ominus, \oplus, \odot, \oslash\}$ has the *north*, *east*, *south*, *west*, *reboot*, *charge*, *analyze*, and *transmit* actions. The transition and reward functions T and R are designed for the analysis task of the task process Υ.

B. Safety Processes

We consider each safety process, $\theta = \langle \bar{S}, \bar{P}, \bar{T}, \phi, \psi \rangle \in \Theta$, designed to address a safety concern of the planetary rover. Intuitively, each safety process has information about its safety concern and can adjust the action performed by the task process by changing its wheel rotation rate (i.e., *speed*) and its steering (i.e., *direction*).

Formally, each safety process $\theta \in \Theta$ has a set of states $\bar{S}_\theta$ that describe the safety concern but the same set of parameters $\bar{P} = \bar{P}_1 \times \bar{P}_2$ with parameter factors $\bar{P}_1$ and $\bar{P}_2$: the wheel rotation rate parameter factor $\bar{P}_1 = \{\Downarrow, \Uparrow, \Diamond, \varnothing\}$ has the *slow*, *speed*, and *stop* actions while the steering parameter factor $\bar{P}_2 = \{\Leftarrow, \Rightarrow, \varnothing\}$ has the *shift left* and *shift right* actions (with the $\varnothing$ symbol that indicates no adjustment). The transition, severity, and interference functions $\bar{T}_\theta$, ϕ_θ, and ψ_θ are designed for the safety concern of the safety process $\theta \in \Theta$. We describe each safety process below.

1) Crevices: The process, $\theta_c = \langle \bar{S}_c, \cdot, \cdot, \cdot, \cdot \rangle$, monitors for crevices to prevent the planetary rover from inhibiting the movement of its wheels. The set of states $\bar{S}_c = F^1_c \times F^2_c \times F^3_c \times F^4_c$ crosses the horizontal rover position relative to the crevice $F^1_c = \{\text{NONE}, \text{APPROACHING}, \text{AT}\}$, the vertical rover position relative to the crevice $F^2_c = \{\text{NONE}, \text{LEFT}, \text{CENTER}, \text{RIGHT}\}$, the rover speed $F^3_c = \{\text{NONE}, \text{LOW}, \text{NORMAL}, \text{HIGH}\}$, and the rover offset relative to its normal path $F^4_c = \{\text{LEFT}, \text{CENTER}, \text{RIGHT}\}$.

2) Dust Storms: The process, $\theta_d = \langle \bar{S}_d, \cdot, \cdot, \cdot, \cdot \rangle$, monitors for dust storms to prevent the planetary rover from damaging its sensitive sensors. The set of states $\bar{S}_d = F^1_d \times F^2_d$ crosses the dust storm density $F^1_d = \{1, 2, \ldots, J\}$ with a limit J and the rover mode $F^2_d = \{\text{ISAWAKE}, \text{ISSLEEPING}\}$.

3) Rough Terrain: The process, $\theta_r = \langle \bar{S}_r, \cdot, \cdot, \cdot, \cdot \rangle$, monitors for rough terrain to prevent the planetary rover from damaging its wheels. The set of states $\bar{S}_r = F^1_r \times F^2_r \times F^3_r$ crosses the horizontal rover position relative to the crevice $F^1_r = \{\text{NONE}, \text{APPROACHING}, \text{AT}\}$, the rover speed $F^2_r = \{\text{NONE}, \text{LOW}, \text{NORMAL}, \text{HIGH}\}$, and the terrain roughness $F^3_r = \{1, 2, \ldots, K\}$ with a limit K.

V. Experiments

We demonstrate that our approach is effective in simulation by comparing a standard planetary rover to different safety metareasoning planetary rovers. The standard planetary rover r_0 does not have any safety metareasoning while each safety metareasoning planetary rover $r_{i>0}$ has a growing set of safety processes: $\Theta^{r_0} = \{\}$, $\Theta^{r_1} = \{\theta_c\}$, $\Theta^{r_2} = \{\theta_c, \theta_d\}$, and $\Theta^{r_3} = \{\theta_c, \theta_d, \theta_r\}$.

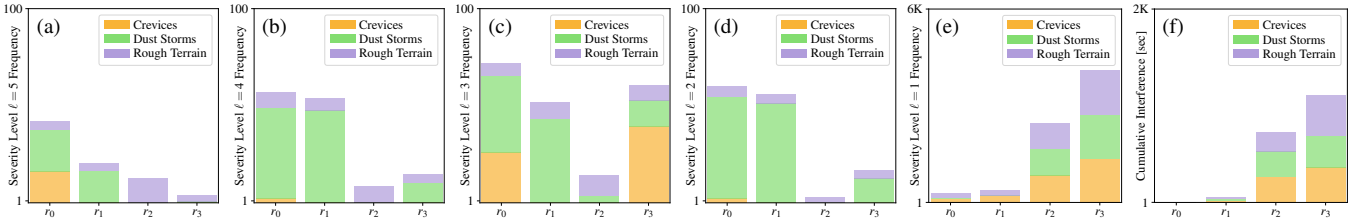

Fig. 3. The performance of each planetary rover for the severity levels and the interference starting with no safety processes and ending with all safety processes where (a) to (d) have a limit of 100 as unsafe operation is rare, (e) has a limit of 6000 as safe operation is common, and (f) has a limit of 2000.

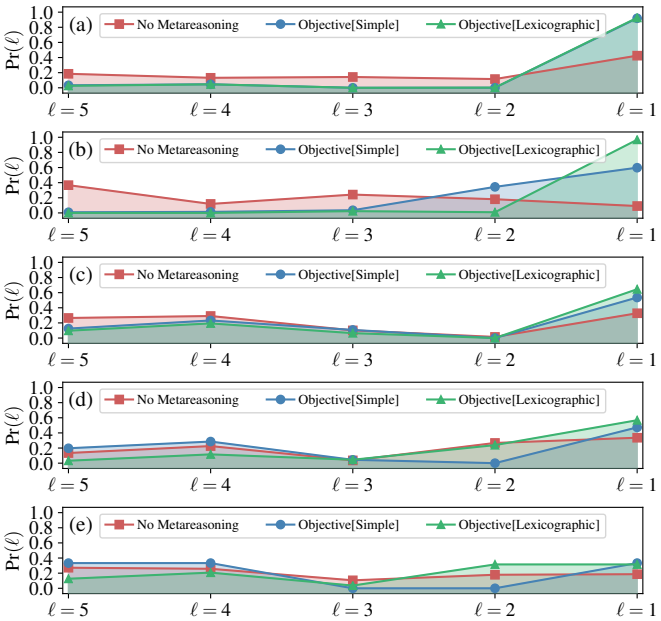

Fig. 4. The severity level probability distributions when different combinations of safety concerns occur across every simulation: (a) is for either no safety concern or isolated crevices, dust storms, and rough terrain, (b) is for simultaneous crevices and dust storms, (c) is for simultaneous crevices and rough terrain, (d) is for simultaneous dust storms and rough terrain, and (e) is for simultaneous crevices, dust storms, and rough terrain.

Each planetary rover must complete the analysis task while addressing the safety concerns that can occur stochastically either in isolation or simultaneously during 50 simulations. For the analysis task, the internal components of the planetary rover begin with a battery level $b = M = 10$, health statuses $h_1 = h_2 = \text{NOMINAL}$, and an objective report $o = (\text{FALSE}, \text{FALSE})$ while the region of the planet has $|I| = 2$ points of interest in an $n = 10$ by $m = 10$ grid such that each cell has weather of a type $w = \text{LIGHT}$ with 0.8 probability or $w = \text{DARK}$ with 0.2 probability. For dust storms, the dust storm density limit is $J = 10$. For rough terrain, the terrain roughness limit is $K = 10$.

Figure 3 shows that our proposed approach is effective in simulation. In Figure 3(a) and (e), at the highest and lowest severity levels, the severity level 5 frequency decreases while the severity level 1 frequency increases from r_0 to r_3 as expected. In Figure 3(b), (c), and (d), at the middle severity levels, the severity level 4, 3, and 2 frequencies remain roughly equal or decrease from r_0 to r_2 but then increase at r_3. This is because the severity level 4, 3, and 2 frequencies for crevices and dust storms must increase to decrease the severity level 5 frequency for rough terrain because a lower severity level is strictly preferred to a higher severity level

TABLE I

A COMPARISON OF A NAIVE APPROACH TO OUR PROPOSED APPROACH.

Capabilities	Size[Naive]	Size[Proposed]	Overhead[Proposed] (s)
Analysis Task	16000	16000	$4.23\text{e}{-7} \pm 1.75\text{e}{-8}$
+ Crevices	+ 2288000	+ 144	$6.63\text{e}{-5} \pm 2.53\text{e}{-7}$
+ Dust Storms	+ 43776000	+ 20	$7.97\text{e}{-5} \pm 1.16\text{e}{-7}$
+ Rough Terrain	+ 5483520000	+ 120	$1.08\text{e}{-4} \pm 2.87\text{e}{-7}$

due to the lexicographic objective function. In Figure 3(f), the cumulative interference increases from r_0 to r_3 as expected. This is because the interference must increase to shift the severity level frequencies from the severity level 5 to 1 but only as much as necessary due to the lexicographic objective function. Overall, the system optimizes the severity of its safety concerns and the interference to its task.

Figure 4 compares our proposed approach with the *lexicographic* objective function to a *simple* objective function for arbitration. The simple objective function always addresses safety concerns sequentially and independently: it first addresses a crevice (if any), then a dust storm (if any), and finally rough terrain (if any) *without* reasoning about how addressing one safety concern could impact other safety concerns or how addressing multiple safety concerns could be performed simultaneously. For each figure, the lexicographic objective function exhibits severity level probabilities that encourage low severity levels but discourage high severity levels compared to the simple objective function.

Table I compares our proposed approach to a *naive* approach that would use a monolithic MDP with every feature of the analysis task and each safety process. The naive approach, however, is intractable given the complexity of its state space and action space. Generally, as the agent becomes capable of addressing each safety concern by including the set of states for each safety process, the naive approach grows multiplicatively while our proposed approach grows additively with negligible overhead for arbitration.

VI. CONCLUSION

We introduce a disciplined, decision-theoretic metareasoning approach to safe decision making in autonomous systems that optimizes the severity of its safety concerns and the interference to its task. By decoupling a safety metareasoning system into a task process and safety processes, our approach offers a key benefit: it provides a framework for autonomous systems to complete a task while addressing safety concerns in a way that avoids a monolithic decision-making model that is often not only intractable but also infeasible to build correctly. Future work will explore complex, general-purpose safety processes for a variety of common safety concerns.

REFERENCES

[1] J. Svegliato, K. H. Wray, S. J. Witwicki, J. Biswas, and S. Zilberstein, "Belief space metareasoning for exception recovery," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2019.

[2] C. Basich, J. Svegliato, K. H. Wray, S. Witwicki, J. Biswas, and S. Zilberstein, "Learning to optimize autonomy in competence-aware systems," in *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, 2020.

[3] C. Basich, J. Svegliato, A. Beach, S. Zilberstein, K. H. Wray, and S. J. Witwicki, "Improving competence via iterative state space refinement," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[4] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *arXiv:1606.06565*, 2016.

[5] S. Saisubramanian, E. Kamar, and S. Zilberstein, "A multi-objective approach to mitigate negative side effects," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2020.

[6] S. Saisubramanian, S. C. Roberts, and S. Zilberstein, "Understanding user attitudes towards negative side effects of AI systems," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

[7] S. Zhang, E. H. Durfee, and S. P. Singh, "Minimax-regret querying on side effects for safe optimality in factored MDPs," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018.

[8] R. C. Arkin, "Governing lethal behavior: Embedding ethics in a hybrid deliberative/reactive robot architecture," in *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction*, 2008.

[9] J. Shim, R. Arkin, and M. Pettinatti, "An intervening ethical governor for a robot mediator in patient-caregiver relationship," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017.

[10] D. Kasenberg and M. Scheutz, "Norm conflict resolution in stochastic domains," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[11] J. Svegliato, S. Nashed, and S. Zilberstein, "An integrated approach to moral autonomous systems," in *Proceedings of the 24th European Conference on Artificial Intelligence*, 2020.

[12] J. Svegliato, S. B. Nashed, and S. Zilberstein, "Ethically compliant sequential decision making," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

[13] S. B. Nashed, J. Svegliato, and S. Zilberstein, "Ethically compliant planning within moral communities," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2021.

[14] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," *Proceedings of the 30th Conference on Neural Information Processing Systems*, vol. 29, 2016.

[15] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, and A. Dragan, "Inverse reward design," *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

[16] J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg, "Scalable agent alignment via reward modeling: a research direction," *arXiv preprint arXiv:1811.07871*, 2018.

[17] R. Dearden, T. Willeke, R. Simmons, V. Verma, F. Hutter, and S. Thrun, "Real-time fault detection and situational awareness for rovers," in *IEEE Aerospace Conference*, 2004.

[18] V. Verma, G. Gordon, R. Simmons, and S. Thrun, "Particle filters for rover fault diagnosis," *IEEE RA Magazine*, 2004.

[19] J. P. Mendoza, M. Veloso, and R. Simmons, "Mobile robot fault detection based on redundant information statistics," in *Proceedings of the IROS Workshop on Safety in Human-Robot Coexistence and Interaction*, 2012.

[20] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Fault detection in a mobile robot using multiple-model estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1998.

[21] P. Goel, G. Dedeoglu, S. I. Roumeliotis, and G. S. Sukhatme, "Fault detection in a mobile robot using a neural network," in *Proceedings of the International Conference on Robotics and Automation*, 2000.

[22] A. S. Manne, "Linear programming and sequential decisions," *Management Science*, 1960.

[23] R. Bellman, "Dynamic programming," in *Science*. American Association for the Advancement of Science, 1966.

[24] J. Svegliato, K. H. Wray, and S. Zilberstein, "Meta-level control of anytime algorithms with online performance prediction," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018.

[25] J. Svegliato, P. Sharma, and S. Zilberstein, "A model-free approach to meta-level control of anytime algorithms," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020.

[26] J. Svegliato, S. J. Witwicki, K. H. Wray, and S. Zilberstein, "Introspective autonomous vehicle operational management," U.S. Patent 10,649,453, May 2020.

[27] A. Bhatia, J. Svegliato, S. B. Nashed, and S. Zilberstein, "Tuning the hyperparameters of anytime planning: A metareasoning approach with deep reinforcement learning," in *Proceedings of the 32nd International Conference on Automated Planning and Scheduling*, 2021.

[28] J. Svegliato, "Metareasoning for planning and execution in autonomous systems," Ph.D. dissertation, University of Massachusetts Amherst, 2022.

[29] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[30] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, 1998.

[31] A. Kolobov and D. Weld, "A theory of goal-oriented MDPs with dead ends," *arXiv preprint arXiv:1210.4875*, 2012.

[32] F. Geißer, D. Speck, and T. Keller, "An analysis of the probabilistic track of the IPC 2018," in *Proceedings of the ICAPS 2019 Workshop on the International Planning Competition*, 2019.