AIAA SciTech Forum
11–15 & 19–21 January 2021, VIRTUAL EVENT
AIAA Scitech 2021 Forum

10.2514/6.2021-1469

Decentralized Context-Based On-Board Planning for Earth Observation Missions

Antoni Viros i Martin*
Texas A&M University, College Station, Texas, 77843, USA

Kewei Cheng† and Zhaoliang Zheng‡
University of California, Los Angeles, Los Angeles, California, 90095, USA

Amy Fang§
Cornell University, Ithaca, New York, 14853, USA

Hadas Kress-Gazit¶
Cornell University, Ithaca, New York, 14853, USA

Ankur Mehta‖ and Yizhou Sun**
University of California, Los Angeles, Los Angeles, California, 90095, USA

Daniel Selva††
Texas A&M University, College Station, Texas, 77843, USA

Currently, the observation plans for most Earth observation satellites are static and periodic, and centrally uploaded from ground control. When there is an event of interest that requires changing the plan, new plans are manually patched and uploaded to the spacecraft. This approach is limited in that: 1) it may lead to missing short-lived phenomena, and 2) it does not scale well to very large constellations of satellites. New mission concepts are starting to leverage Artificial Intelligence to make autonomous planning decisions in reaction to short-lived events, which has spurred interest in on board autonomous scheduling. In addition, emerging mission concepts rely more on distributed systems (clusters, constellations) of heterogeneous satellites, in some cases with the ability to communicate through cross-links to coordinate their operation. Motivated by these recent trends towards increased system complexity, decentralization and on-board autonomy and advances in sensor fusion theory, this paper explores a significantly different observing paradigm and corresponding problem in which satellites receive a mission (observations of a target region with a certain set of required attributes and time constraints) from mission control and autonomously decide whether or not they can and should participate in such a mission based on their current state and contextual information. Specifically, we propose a new approach based on three main steps: first, each satellite determines if it *can* participate in the mission given the mission specification and its sensor characteristics by reasoning over a knowledge graph; second, each satellite assesses if it *should* participate in the mission by estimating the probability that it can measure the observable of interest using a decentralized Kalman filter. Third, a central node can then verify the emerging plan using a probabilistic temporal logic and synthesize a better plan if desired. The new approach is described in detail and evaluated for a case study concerning the detection of volcano eruptions. In order to validate our methodology, we compare a set of teams generated by our method for the problem of volcano eruption detection to a team that is currently being used at NASA to

*Graduate Research Assitant, Department of Aerospace Engineering, 3141 TAMU, College Station, Texas, 77843, AIAA Student Member
†Graduate Research Assistant, Department of Computer Science, 404 Westwood Plaza, Engineering VI, Los Angeles, CA 90095
‡Graduate Student Researcher, Electrical and Computer Engineering Department, 420 Westwood Plaza (Box 951594), Los Angeles, CA 90095
§Graduate Research Assistant, Sibley School of Mechanical and Aerospace Engineering, 130 Upson Hall, Ithaca, New York 14853
¶Associate Professor, Sibley School of Mechanical and Aerospace Engineering, 130 Upson Hall, Ithaca, New York 14853
‖Assistant Professor, Electrical and Computer Engineering Department, 420 Westwood Plaza (Box 951594), Los Angeles, CA 90095
**Associate Professor, Department of Computer Science, 404 Westwood Plaza, Engineering VI, Los Angeles, CA 90095
††Assistant Professor, Department of Aerospace Engineering, 3141 TAMU, College Station, Texas, 77843, AIAA Senior Member

perform the same task, with positive results. Finally, we discuss the limitations of the current method and outline a future path for the method as a whole as well as the individual steps.

I. Introduction

This paper introduces a new decentralized methodology for autonomous on-board planning of Earth Observation (EO) missions using contextual information. As requirements for EO missions grow tighter, re-purposing available EO satellites currently on orbit is increasingly considered as a valid alternative to launching new missions, especially for missions that require high revisit times only attainable through the use of constellations [1]. For example, NASA's Technology Taxonomy* (a compendium of technologies that are of interest to advancing NASA's mission) identifies technologies that foster reasoning, acting, collaboration, and interaction as priorities for development in space autonomous systems in the next years.

For the most part, current EO satellites are similar to silos as far as planning goes: they have a simple static imaging concept (e.g., cross-track scanning) that implicitly defines an observation schedule before the beginning of the mission and that schedule never or rarely changes. Some newer imaging satellites actually have pointing capabilities, like the Sentinel constellation [2], and they can modify those static plans to capture events of interest. It is quite rare to find systems for which the schedule is fully defined in near real time, but when this happens (e.g., the Planet constellation [3]) the scheduling problem is solved on the ground and plans uploaded to individual satellites. In these few cases the system is always homogeneous and centrally owned and operated by one actor. Many approaches to scheduling and planning for EO missions have been developed assuming this centralized, on-ground paradigm, including approaches based on Constraint Satisfaction [4], Genetic Algorithms [5], or Integer Programming [6].

One limitation of the ground-based planning approach is that it potentially misses short-lived events due to the time incurred between the observation and the data processing on the ground (which includes the time from observation to downlink plus the time for data distribution and processing). Stochastic urgent events require fast revisit times, but the average revisit time for many of the most commonly used satellites (which are in Low Earth Orbit) is on the order of a day at best (such as the observations of MODIS from the Aqua and Terra satellites) and up to two weeks (e.g. Landsat satellites), and that is only if the satellites have no other constraints, such as VNIR sensors requiring cloud-free observations. This revisit time may be too long for short-lived or short-notice events such as cloud formation or volcano eruptions.

A limitation of the centralized planning approach it does not scale well to very large constellations of satellites (which are gaining popularity) due to requiring expert input. These experts have to be knowledgeable on contextual information about the mission and available assets to make a decision, and that leaves no time for evaluating many different options, which may lead to a solution that is not optimal.

Both of these limitations are particularly problematic given the trends in EO system architectures. Experts in the field have postulated for decades that the future of EO is the Sensor Web [7], i.e., decentralized networks of intelligent nodes carrying heterogeneous sensors with on-board decision making and cross-link capabilities. However, nothing close to this has been actually flown. In terms of a having a large constellation of heterogeneous systems actually flown, QB50 [8] is the closest system. QB50 is a constellation of CubeSats designed and launched by different universities with heterogeneous and complementary sensors, but it lacks the advanced on-board decision making and full cross-link capabilities. These technologies however have been demonstrated independently – satellite cross-links by Iridium for example [9] and on-board decision making by EO-1 [10].

Another proposed (but not demonstrated) concept that illustrates the trend towards decentralization and on-board decision-making is that of federated systems [1], in which satellites owned and operated by different organizations come together opportunistically and temporarily to work on emerging tasks.

Our approach is particularly suited for such future EO system concepts, as the different actors involved in a federation or large distributed system like QB50 may have used different onboard "dictionaries" to define and reason about their satellites' capabilities. Our approach's context mining and reasoning are suited to deal with different sources of data to harmonize them and get some level of interoperability between these satellites.

All these new approaches to EO –the Sensor Web, federated systems, the QB50 constellation– require new decentralized planning algorithms. Other than the approach described in this paper, in recent years there has been a constant stream of research in on-board scheduling for EO satellites. In [11], Li et al. introduce an autonomous online approach to satellite scheduling that can handle urgent tasks arriving stochastically during the planning by using two heuristics to decide when to schedule and how to do so. In more recent work, Li [12] expands their work to encompass a

*https://www.nasa.gov/offices/oct/taxonomy/index.html whole distributed satellite system, and bases their algorithm on the asynchronous version of the consensus-based bundle algorithm (ACBBA) [13]. Another example, also based on CBBA, is the approach by Gallud and Selva described in [14] for decentralized planning in distributed and federated EO systems.

As an example of a representative problem and corresponding distributed EO system, consider the case of detecting volcano eruptions. NASA has developed two parallel approaches for this purpose. On the one hand, it has been using data from both the Aqua and Terra satellites's MODIS sensor to create MODVOLC[15], a detector for eruptions around the Earth with a revisit frequency between 24 and 48h, which may not be fast enough in the case of explosive eruptions. The other, more recent, approach is described in detail in [16]. It combines in-situ and space observations to monitor when an eruption happens and trigger special observations for some satellites and ground assets. In the case of ESA, they developed the Geohazards Exploitation Platform (GEP)[†], which combines images and radar information from the Sentinel 1 and 2 constellations to detect natural disasters, including volcano eruptions. In the three cases, pointing satellites to take extra images of a region of interest requires a complicated process involving humans and approval processes on the ground.

In this paper, we propose a new approach for the decentralized planning problem for heterogeneous, distributed (and potentially federated) EO systems. In this approach, agents independently choose whether and how to respond to a new mission through the use of contextual information. A preliminary offline step of our approach is the mining of such contextual information. Through this paper, we use an extensive definition of contextual information, including facts ranging from the capabilities of a satellite and its sensors such as accuracies, power, image resolution, fields of view, and other characteristics, as well as information about their current status including orbital information and whether sensors are operational or not. This contextual information also includes knowledge about the physics of the Earth system and remote sensing, such as how different so-called Level-1 measurements (e.g., TIR radiances or radar back-scatter cross-sections in L-band) relate to geophysical parameters or Level-2 products (e.g., land surface temperature or soil moisture) and the relation between a mission specification and a set of geophysical parameters to observe. This context could optionally include information about other satellites and their capabilities. Going back to the volcano eruption example, this means the context has information on volcano eruptions generating heat, land displacements, and ash plumes, and we also know that Thermal Infrared (TIR) and Short-Wave Infrared (SWIR) radiance can be used to measure temperature at different ranges, and SAR interferometry can provide ground surface motions (gradients of velocity vectors). Note the wealth of information that is fed to the planning algorithm. In comparison, the approaches mentioned before only take into account a much more narrowly defined state of the satellite as needed to schedule their missions. This comprehensive context information is intended to help with planning by eliminating the need for an EO expert at different steps of the process. While a classic approach requires an expert to realize that TIR sensors can be used to measure the temperature increase in the surface of a volcano during an eruption and then consider the available TIR sensors, our proposed method can derive this piece of information by itself and conduct those steps autonomously. Once this is done, the algorithm can use each satellite and sensor status information to decide what are the best out of all these to participate in a mission to monitor a specific volcano. In order to extract all the necessary information about sensors, observables, etc, we have used unstructured text mining techniques to fill a probabilistic knowledge graph (KG), which is a set of tuples consisting of a fact represented as a triplet (concept C_1, relation R, concept C_2) and a probability p of that fact being true. This KG is then coupled with a mixed probabilistic- and logic-based reasoning system in order to infer more relations and reason about which satellites can participate in a given mission. Since such a general problem has never been considered before in EO, no other approach in the literature uses a similar reasoning system to start the planing procedure.

Once we have the contextual information for each agent, our approach has three steps, each adding significant novelty to the way scheduling is done for EO systems: first, each satellite decides whether it *can* participate in the mission by comparing its capabilities with the mission specification (e.g., a satellite with a VNIR or TIR imager can participate in a volcano detection mission) and checking if it has potential visibility of the target region in the specified period of time. This step is done by reasoning over a knowledge graph (partially mined from scientific papers) containing relations between different sensor types, geophysical parameters and observables. While this first step provides a very computationally efficient down-selection of the agents that need to consider the mission, it is essentially a binary check that only considers the capabilities of the satellite as opposed to the expected performance. Hence, in the second step, each satellite that has decided it can participate rates its potential to add information and thus value to the mission by considering more detailed sensor specifications (e.g., sensor accuracy) and the specific processes involved, in order to decide if it *should* participate. In the literature, these scores are usually calculated semi-qualitatively. For example, in

[†]https://geohazards-tep.eu/

the VASSAR methodology [17], scores are computed through a set of rules that take into account sensor characteristics such as spatial resolution, revisit time, accuracy, etc. Often, this number comes from subjective numbers based on expert judgment. Rather than relying on subjective assessments of the value of different combinations of parameters, our approach is based on the physics and mathematics of sensing. Because computing a full error budget for the parameters of interest onboard is impossible based on the information available, we propose a new simple but powerful theoretical framework to estimate the uncertainty in the measurements of each sensor based on a decentralized Kalman filter. In this paper, we consider the case of an event detection mission (specifically volcanic eruption detection) as opposed to other types of missions. Hence, the result of the Kalman filter is fed to a classical binary hypothesis testing framework to assess the probability that the sensor can detect the event of interest. In a completely decentralized system, each satellite could simply decide to act based on the output of this step. However, in practice, there is likely value in considering some coordination between the satellites. In this paper, we consider the case where a central node synthesizes a teaming plan based on the input from each sensor and formally verifies that it can satisfy the mission using probabilistic temporal logic. While the approach does do some search for a good team, it does not find the optimal team as it is a computationally intractable problem. The formal methods approach to the verification of the decentralized plan based on probabilistic model checking is very new for EO and provides a rigorous theoretical framework to synethesize and reason about plans including complex constraints, e.g. related to scarce on board resources. Our method is applicable for space-based observation of any geophysical parameter, although it is most useful for remote sensing of short-lived or short-notice processes and phenomena by constellations of satellites with heterogeneous sensors (e.g. QB 50) or federated systems.

More generally, the problem we are trying to solve is formulated as follows: given a **mission** specified by 1) a **geophysical parameter** of the Earth to observe such as sea surface height or soil moisture, or an **event** such as a volcano eruption, 2) a **region** (e.g. the Arctic, the Mediterranean basin, or the Pacific Ring of Fire), and 3) some **time-related requirements** (mission duration of a week or month, revisit time of two measurements a day, or a maximum gap between measurements of 24 hours), find **a selection of teams of satellites** that will perform the required observations to successfully carry out the mission. Optionally, the mission statement can also include other parameters such as spatial resolution, accuracy, etc. Of note, relevant requirements for many parameters can be readily obtained from the online World Meteorological Organization (WMO) OSCAR database[‡].

The main performance metric in this paper is in the form of a probability of success in the mission. Since teams with more satellites have higher probabilities of success, the output of the system is a Pareto frontier of teams generated by our decentralized approach, with a probability of the mission succeeding for each team vs the number of satellites involved. This probability can be calculated a priori without seeing any dataset. If a real dataset is provided, a probability for each of the teams successfully carrying out the mission in the context of the data can be computed for verification.

The set of available agents can be the set of all Earth observing satellites currently in space, which can be extracted from the CEOS database[§]. This database contains information about all the current and planned Earth observation missions and their sensors.

For example, if the mission is to measure soil moisture in the Mediterranean basin daily for one month, one would expect that the Soil Moisture and Ocean Salinity (SMOS) and Soil Moisture Active Passive (SMAP) satellites would both volunteer to participate in the mission, but more sensors (e.g., C-band radar from Sentinel) may be required to achieve the desired mission attributes. In order to compare against a team that is already in the literature to validate our approach, we formulate a simple but general and relevant test problem consisting of detecting a volcano eruption based on remote sensing of land surface temperature, aerosol plumes (especially SO2), and land surface displacements in one out of the 10 most active volcanic areas on Earth, but the problem is readily generalized to any other event that can be linked to one or more geophysical parameters.

In order to ensure our planning algorithm works, we compare our set of teams to the team used in [16] to perform exactly the same task. Their team is comprised of the following satellites: Aqua, Terra, the Sentinel 1 constellation, the Sentinel 2 constellation, and all active GOES satellites. Given that a similar metric to the probabilities outputted by our algorithm for that team is unknown, we have processed the team through the same pipeline as our teams in order to ensure a fair comparison.

[‡]https://www.wmo-sat.info/oscar/

[§]http://database.eohandbook.com/

II. Methodology

A. Overview

An outline of the system can be seen in Figure 1. The outline described in the flowchart is the approach with a centralized decision node (represented by the optimization algorithm). In the case of the decentralized approach, the optimization algorithm changes to a threshold-based approach where each satellite decides to participate based on its own perceived usefulness to the mission. The next subsections describe the subsystems in the flowchart, as well as the teaming algorithms.

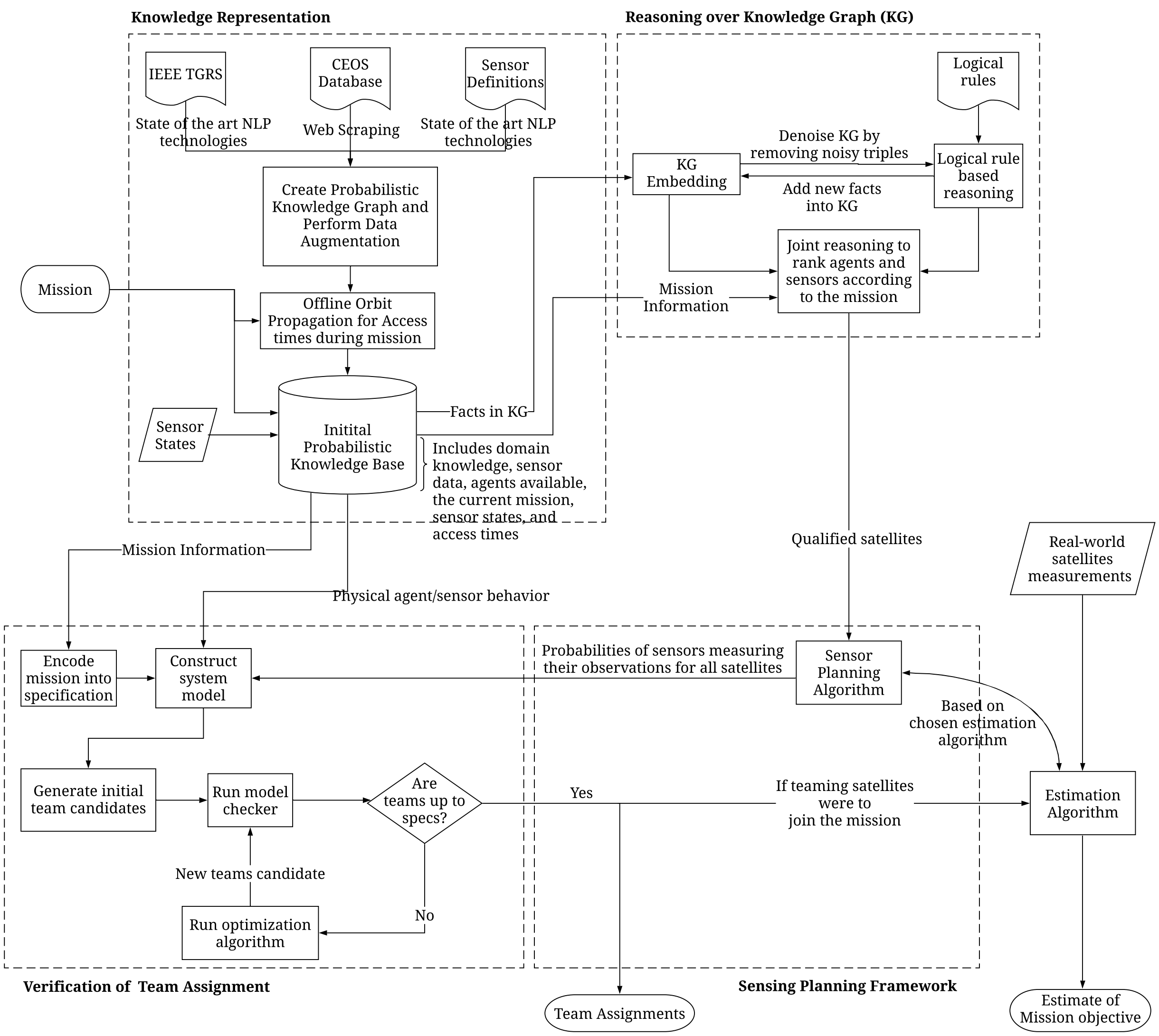


Fig. 1 System flowchart, showcasing the interactions between the components in our approach

B. Knowledge representation

The knowledge representation framework requires different types of knowledge: 1) domain knowledge, which includes knowledge about: a) events of interest and related observable physical phenomena (e.g., volcano eruptions generate heat and SO_2 aerosols), and b) sensor types and their capabilities to measure various geophysical parameters (e.g., a short-wave infrared sensor can measure aerosols, thermal infrared sensors can measure heat); 2) knowledge about the mission at hand (e.g., to detect volcano eruptions in Hawaii); 3) knowledge about the agents and their sensors that are available to perform missions (e.g., a particular agent can measure TIR radiance with 5% accuracy); 4) knowledge about the state of each agent (e.g., is sensor operational, is the target in visibility of the sensor).

These different types of knowledge can all be represented with a probabilistic knowledge base, as seen in Figure 2. Each agent has a copy of this knowledge base, which can either contain only relevant information to the agent or all the information for the system. Part of the knowledge is intrinsically probabilistic, as seen in the figure.

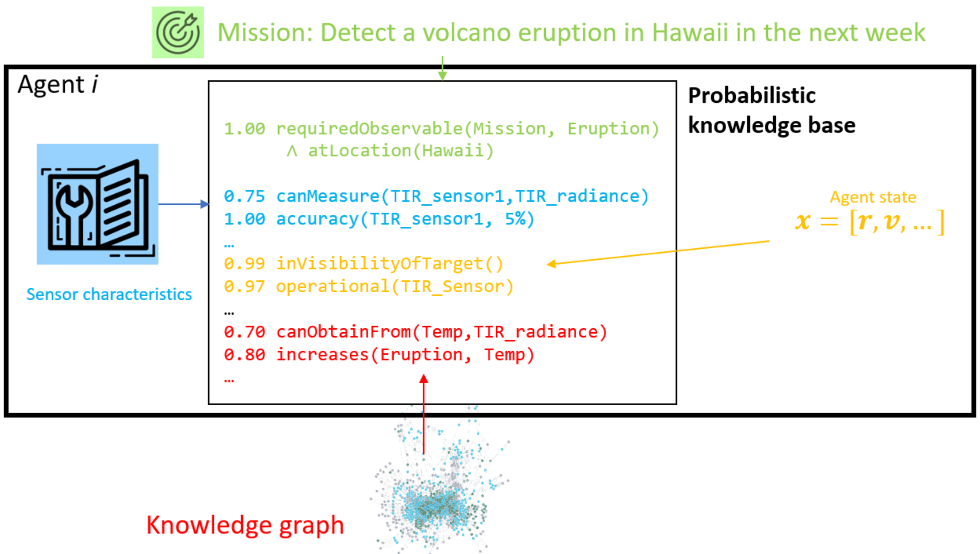


Fig. 2 Different types of knowledge in the probabilistic knowledge base, with their sources and their associated uncertainty

While the sources of knowledge for the mission, the agent state, and the sensor characteristics are relatively clear, finding the domain knowledge linking the sensor capabilities and characteristics to the mission at hand is a challenge. The basic idea is that one must find a path in the KG from the geophysical parameters observable by the sensor to the target observable event required by the mission. This path may in general require multiple steps (e.g. volcano eruption →heat →thermal infrared (TIR) radiation →TIR sensor). If the path already exists in the KG, then one can proceed with inference. However, if such a path does not exist in the KG (e.g., the agent does not currently know that heat is related to TIR radiation), the agent will then attempt to find new information linking those entities from other existing corpuses. In the case an Internet connection is available, the agent will source the information from there. This being said, most satellites in space lack access to the Internet, so they might have to look for such information in local copies stored in each agent. This requires extracting information from unstructured text sources such as databases of papers, using techniques such as Named Entity Recognition (NER) [18] and Relation Extraction (RE) [19]. For the approach described here, we have built a Natural Language Pipeline (NLP) to extract the links between different observations, sensors, and physical phenomena from papers published in the Transactions on Geoscience and Remote Sensing journal[¶]. Specifically, our pipeline includes an NER pipe based on the Scibert Transformer [20], followed by an Entity Linker (EL) to match the extracted entities to out KG. The EL is based on the models in SpaCy 2 [21], and retrained for our KG. The last part of the pipeline is an RE engine that we have built for this project that creates new relationships in the KG between the entities that have been found in the previous steps.

Two longer term research challenges related to both knowledge representation and reasoning in our system are: 1) modeling the different types of uncertainty that exist in the problem, and 2) how the knowledge and its uncertainty change over time.

We define two sources of uncertainty. It may be aleatory (e.g., random noise in the sensor) or epistemic (e.g., lack of information about a sensor, observable, mission, or relations between them). Aleatory uncertainty can be modeled as usual with probability theory. However, for epistemic uncertainty, KGs lack the capacity to easily represent uncertain

[¶]https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=36 facts. We are exploring approaches combining probability and logic, such as Probabilistic Soft Logic [22] and Markov Logic Networks [23]. In Markov Logic Networks (MLN), rules can have weights related to the probability of the rule being true, but the evidence is still Boolean. Soft logic approaches allow for uncertain evidence. Regardless of the implementation, the requirement is to have a probabilistic knowledge base where both the evidence and rules have an associated probability of being true. How we obtain these probabilities for the case of epistemic uncertainty is another challenge that can be addressed, for example, with embedding-based methods, which will be explained in more detail in the next section.

In the case of the time dependency of both knowledge and uncertainty, we have to take into account all the different time-dependent sources of information. For example, the mission might change at any time, as well as the agent state, with sensors being taken down or up depending on system status. While the sensor characteristics are relatively stable, the domain knowledge is continuously being updated with the publication of new information, and a successful system will require keeping up to date with the latest advances in the field in order to stay competitive. Such changes might add new relationships to the KG or change the uncertainty on existing ones.

C. Reasoning over KG

Notation A knowledge graph, denoted by $\mathcal{G} = \{E, R, O\}$, consists of a set of entities E, a set of relations R, and a set of observed facts O. Each fact in O is represented by a triple (e_i, r_k, e_j), where $e_i \in E$, $e_j \in E$, and $r_k \in R$ denote subject entity, object entity, and relation, respectively.

The KG reasoning module is responsible for deciding whether a satellite can participate in a given mission. Given a KG containing both common sense knowledge about the field and knowledge about the sensors we have, and a set of logical rules specified by domain experts, we aim to predict which sensors can participate in the mission by leveraging the relations between geophysical parameters observable by the sensors and the objective events required by the mission.

There are two main directions for solving the posed problem, which is also known as KG inference: logical rule reasoning and knowledge graph embeddings (KGE). Traditional logical inference aims to find unobserved facts given a set of rules and the currently observed facts, which results in maximizing the number of rules that can be satisfied; on the other hand, KGE aims to capture the similarity of entities by embedding entities and relations into continuous low-dimensional vectors. With the learned embeddings, the KGE approach is able to compute a score for each triple in a KG that indicates the plausibility of the triple. We denote the score of a triple (e_i, r_k, e_j) computed by KGE as $f_{r_k}(e_i, e_j)$. Logical rules and KGE can mutually enhance the reasoning ability of each other in an iterative process [24]. To better model the interactions between embedding learning and logical inference, we will use a unified framework to integrate embeddings and First Order Logic Horn rules in an iterative manner called UniKER [25] developed by authors Cheng and Sun, referred as Reasoning over Knowledge Graph in Figure 1. In particular, UniKER shows that the knowledge contained in Horn rules can be exploited and completely transferred them into the embeddings without the need for sampling ground rules or hidden triples in KGs. Additionally, UniKER can tolerate erroneous data and thus shows robustness to noise (extraneous facts) and errors (incorrect facts) in the KGs, which previous methods cannot cope with [26, 27].

The basic idea of the algorithm is as follows: we leverage the reasoning power of Horn rules by deriving a satisfying truth assignment $\mathbf{v}_H^{T\,*}$ and $\mathbf{v}_H^{F\,*}$ by a forward chaining algorithm, where $\mathbf{v}_H^{T\,*} = \{r_k(e_i, e_j) = 1 \mid r_k(e_i, e_j) \in \mathbf{v}_H\}$ and $\mathbf{v}_H^{F\,*} = \{r_k(e_i, e_j) = 0 \mid r_k(e_i, e_j) \in \mathbf{v}_H\}$. Knowledge contained in Horn rules is guaranteed to be fully exploited by taking $\mathbf{v}_H^{T\,*}$ and $\mathbf{v}_H^{F\,*}$ as guidance to optimize the KGE model. We summarize the iterative learning procedure as follows. We start by training a KGE model over the observed facts O. Next, following the forward chaining algorithm, we derive the triples set $\mathbf{v}_H^{T\,*}$. In particular, each iteration consist of the following steps:

1) Considering that potential noise could affect the performance of forward chaining, we regard θ% triples with lowest prediction scores $f_{r_k}(e_i, e_j)$ as noise and remove them from observed facts O to have a cleaner KG.
2) To relieve the data sparsity issue in real-world KGs, we also include potential useful hidden triples to enhance the reasoning ability of Horn rules. Then, we classify all triples in Δ and add the positive triples to O.
3) After that, by looking for the ground rules whose bodies are satisfied in O, we obtain a subset $\mathbf{v}_H^{T\,i*}$ of the satisfying truth assignment $\mathbf{v}_H^{T\,*}$.
4) To reduce some of the uncertainty brought by logical rules, we utilize the trained KGE model to check the correctness of the inferred triples in $\mathbf{v}_H^{T\,i*}$ and exclude θ% triples with lowest prediction scores $f_{r_k}(e_i, e_j)$ from $\mathbf{v}_H^{T\,i*}$.
5) After that, we continually train the KGE model over $\mathbf{v}_H^{T\,i*}$ and add $\mathbf{v}_H^{T\,i*}$ to KGs by setting $O = O \cup \mathbf{v}_H^{T\,i*}$.

The final triples obtained through UniKER are used to create a list of satellites and sensors within them that can participate in a given mission. This list of satellites is then sent to the Sensor Planning Framework to compute whether they should participate in a mission or not.

D. Sensor Planning Framework

The Sensor Planning Framework is an analytic algorithm to compute an a priori error probability on sensing objectives (e.g., the probability of one sensor successfully detecting an event, or the probability of successfully detecting an event if multiple sensors join the mission) based on the contextual knowledge each agent has. We have developed a sensor planning algorithm to calculate the error probability of each participating sensor based on the specific estimation algorithm chosen. Given agents that decide to participate in the mission, the estimation algorithm computes an estimate of the mission objective (e.g. probability of an event $P(Event_t)$) using the agents' sensor capabilities.

To be more specific, it provides the certainty that our sensors are going to tell the truth about their own observation, i.e. the probability that a sensor successfully detects the corresponding observation of an event given the event happened, as well as the probabilities of a false positive, true negative, and false negative. The sensor planning algorithm assumes the process is linear and estimation of mission objectives are done through the estimation algorithm. In our case, we use a Kalman Filter series estimation algorithm such as the ones in [28] [29] [30], given their track record in satellites and other vehicles. Kalman filters can also be applied to multi-sensor data fusion [31] [32].

Thus, our method takes the following matrices as input for each sensor from the KG: Process Model:=A, B, Observation Model:= H, Covariance of Measurement noise:=R, Covariance of undetermined/initial process noise:=Q. Then, the algorithm outputs the error bound of the estimation algorithm Σ and $P(Detected|Event)$ over time. The sensor teaming planning algorithm is run before observing the real-world data feed. The idea behind this algorithm is to gather information in order to create a team before it commits to taking the measurements for the estimation algorithm, thus avoiding resources and time waste.

Given observation data from a formed team (e.g. temperature, aerosols, and land displacements close to a volcano), the estimation part of the algorithm will compute the optimal estimated states of the sensors (e.g. near-optimal estimate of real land surface temperature). The estimation algorithm takes real-world measurements: observations of sensors z_t, system matrices of the sensors as mentioned in the previous paragraph and confidence factor w_i of each sensor as inputs, and outputs the probability of an event happening over time $P(Event_t)$, but this time with the real data behind it instead of just an apriori estimation.

The formal definition of the problem is given below:

1. Problem statement

Given A, B, H, Q, R of a sensor, we use this particular sensor to observe physical quantities for a long period of time (e.g. t = 100 days). Then, we assume that on the next time step $t + 1$, the sensor outputs either $u_t = 0 \; or \; 1$ (event does not happen or event happens). First, compute the probability that the sensor decides $u_t = 1$ given the fact that $u_t = 1$ (true positive). Then, compute the probability that the sensor decides $u_t = 0$ given the fact that $u_t = 1$ (false positive). Further, compute the probability that the sensor decides $u_t = 0$ given the fact that $u_t = 0$ (true negative). Finally, compute the probability that the agent decides $u_t = 1$ given the fact that $u_t = 0$ (false negative).

2. Problem Clarification

The probability that we calculate here is actually a testing in a binary situation. Assuming that we have two hypotheses:

- $H_0 : u_t = 0$ (no events happen)
- $H_1 : u_t = 1$ (events happen)

There are four probabilities that need to be computed:

- H_1 is true, decide H_1 : given the H_1 is true hypothesis, decide H1 is in force: $Prob\{decideH_1|H_1true\} = P_D$ (probability of detection)
- H_0 is true, decide H_1 : given the H0 is true hypothesis, decide H_1 is in force: $Prob\{decideH_1|H_0true\} = P_{FA}$ (probability of false alarm)
- H_1 is true, decide H_0 : $Prob\{decideH_0 \mid H_1true\} = P_M = 1 - P_D$ (probability of missing)
- H_0 is true, decide H_0: $Prob\{decideH_0 \mid H_0true\} = P_{CR} = 1 - P_{FA}$ (probability of correct rejection)

3. Method and derivation

Consider the system model and the observation model:

- Prior: $x_t \mid z_{0:t}, u_{0:t-1} \sim N(\mu_{t|t}, \Sigma_{t|t})$
- System model: $x_{t+1} = Ax_t + Bu_t + w_t, w_t \sim N(0, Q)$, where $x_{t+1}|x_t \sim N(Ax_t + Bu_t, Q)$
- Observation model: $z_{t+1} = Hx_{t+1} + v_t, v_t \sim N(0, R)$, where $z_t|x_t \sim N(Hx_t, R)$

Using a Kalman filter as a linear quadratic optimal estimator, the Kalman prediction step to estimate state x and its likelihood is defined as follows:

- $\mu_{t+1|t} = A\mu_{t|t} + Bu_t$
- $\Sigma_{t+1|t} = A\Sigma_{t|t}A^T + Q$

Then, we define the update step as follows:

- $\mu_{t+1|t+1} = \mu_{t+1|t} + K_{t+1|t}(z_{t+1} - H\mu_{t+1|t})$
- $\Sigma_{t+1|t+1} = (I - K_{t+1|t}H)\Sigma_{t+1|t}$

where the Kalman gain is: $K_{t+1|t} = \Sigma_{t+1|t}H^T(H\Sigma_{t+1|t}H^T + R)^{-1}$

In the Kalman filter, the objective is to calculate:

$$p_{t+1|t+1}(x) = \frac{p(z_{t+1}|x)p_{t+1|t}(x)}{p(z_{t+1}|z_{0:t}, u_{0:t})}$$

In our method, the objective is to calculate uncertainty of the input in the last time step u_t. More specifically, given a binary candidate input sequences $u_t^j = \{0, 1\}, j \in \{1, 2\}$, evaluate the likelihood of each input over time given only observations $z_{0:t+1}$. Therefore, the error probability function that we try to compute is $pdf(u_t|z_{0:t+1}, x_{0:t}, u_{0:t-1})$, denote it as $pdf(U)$.

$$pdf(u_t|z_{0:t+1}, x_{0:t}, u_{0:t-1}) = pdf(U) = z_{t+1} - HA\mu_{t|t} = z_{t+1} - H\mu_{t+1|t}$$

In the Kalman filter, $pdf(U)$ is also Gaussian under each hypothesis and its distribution is as follows:

$$U(z_{t+1}) \sim \begin{cases} N(\mu_{u^1}, \Sigma_{u^1}) & under \quad H_0 \\ N(\mu_{u^2}, \Sigma_{u^2}) & under \quad H_1 \end{cases}$$

where,

$\mu_{u_t} = \mathbf{E}[z_{t+1} - H\mu_{t+1|t}] = \mathbf{E}[Hx_{t+1} + v_t - H\mu_{t+1|t}] = \mathbf{E}[HA\mu_{t|t} + v_t - H(A\mu_{t|t} + Bu_t)] = \mathbf{E}[v_t + HBu_t] = HB\mathbf{E}[u_t]$

$\Sigma_{u_t} = \mathbf{Var}[z_{t+1} - H\mu_{t+1|t}] \overset{indepen}{=} \mathbf{Var}[z_{t+1}] = \mathbf{Var}[Hx_{t+1} + v_t] = H\Sigma_{t+1|t}H^{-1} + R = H(A\Sigma_{t|t}A^T + Q)H^{-1} + R$

Therefore, the error probabilities P_D, P_{FA}, P_M, P_{CR} can be calculated using the following equations:

$P_D = \int_\lambda^\infty pdf_{H_1}(U) = \int_\lambda^\infty \mathbf{N}(\mu_{u^2}, \Sigma_{u^2})$

$P_{FA} = \int_\lambda^\infty pdf_{H_0}(U) = \int_\lambda^\infty \mathbf{N}(\mu_{u^1}, \Sigma_{u^1})$

$P_M = 1 - P_D$

$P_{CR} = 1 - P_{FA}$

where, λ is when $\mathbf{N}(\mu_{u^2}, \Sigma_{u^2}) = \mathbf{N}(\mu_{u^1}, \Sigma_{u^1})$

For each single sensor that observes a single geophysical property, we can use the above method to calculate their error probabilities respectively. In the case of a sensor with more than one measurement, we assume it can be split into multiple sensors so the former is always valid.

The results of this portion of our approach are used to inform the creation of teams by either the fully decentralized approach or the team synthesis approach described in the following section (II.F).

E. Verification and Synthesis of Team Assignment

Formal verification has been used in a wide range of applications to verify the safety and correctness of systems with respect to given specifications [33] [34] [35]. In our system, the aim of the verification step is to provide information about the likelihood of mission success given a teaming assignment and sensor properties. Individual agents decide whether to participate or not, but a centralized decision needs to be made to ensure that the mission constraints are met at a team level. If it is determined that the mission is likely to fail, the Sensor Planning Framework then assigns a new team of agents.

Alternatively, the centralized node can also create a teaming plan using synthesis. Formal synthesis provides a framework for automatically translating high-level specifications into correct-by-construction controllers. The user is able to reason about the task specification rather than the actual implementation [36]. The aim of the synthesis step is to create an optimal teaming plan given a list of possible agents and their capabilities. This method provides a globally optimal teaming solution.

There are three main steps in performing synthesis for the mission:

1) **Encoding the mission as a formal specification.** Specifications are often written using temporal logic [37], which allows us to reason about requirements and constraints that change over time. In particular, we use probabilistic computation tree logic (PCTL) [38] to write the specification. The specification can also include constraints on the system (e.g. sensor interference).
2) **Creating a system model.** The assigned team is modeled as a Markov Decision Process (MDP). Each state within the MDP contains information about which agents and sensors are used, as well as which observations are measured.
   The uncertainty in the sensor readings are represented as probabilities, which change over time as the Sensor Planning Framework receives more measurements to improve the sensors' state estimations. From these probabilities, we calculate the transitions between states in the MDP. The transitions represent the probabilities that a specific combination of sensors measure the necessary observations are measured at a given timestep.
   Also encoded in the MDP is the physical behavior of the agents. For example, in our use case, the target volcano is only visible to the satellites periodically.
3) **Verifying the decentralized team.** Once the mission specification and system model is created, we use PRISM [39], a probabilistic model checker, to calculate the probability of success of the decentralized teaming plan. If this probability falls below a designated threshold, the verification step requests a new team assignment from the Sensor Planning Framework.
4) **Synthesizing the optimal team.** Optionally, the centralized node can synthesize a globally optimal team from a list of potential agents. To do so, the model checker maximizes the probabilities of the MDP state transitions, while taking into account any constraints given in the specification. If the mission contains multiple objectives (e.g. maximize the probability of mission success while minimizing the number of satellites), PRISM outputs the Pareto front for all viable teams.

F. Teaming Algorithm

The teaming decision, as explained in the introduction, can be done through two approaches, depending on whether centralization of decisions is acceptable. Both of them start with mission control sending a mission to all satellites. Each satellite and sensor comes up with a probability of completing the mission successfully (aka detecting the event in our case), as described in Section II.D. In the decentralized case, if they clear a predetermined threshold they decide to participate. In the centralized approach, a central node receives the probabilities from each satellite and performs a greedy optimization based on the success probability for the chosen team, creating a Pareto front of teams of different sizes and their respective probability of success.

III. Results

In order to validate our complete approach to the planning problem, we compare the teaming results of our approach to the team described in [16]. This comparison can be seen in Figure 3, where each dot is a team with their daily assignment of satellites (e.g., Day 1: Sentinel-1, EO-1, Day 2: GOES-13) (which we will call **Teaming Strategies** going forward), as outputted by the Verification module . The X axis represents the mean number of satellites that have to be used per day, and the Y axis is the maximum computed probability of mission success based off the results of our system. In order to ensure that the scores are fair to each team, each dot is computed as the average of the teaming strategy applied to 50 random missions generated by a Monte-Carlo approach. Thus, the blue dots represent the best teaming strategies by tasking the whole Earth Observation constellation currently on orbit to figure out a plan for our random missions, while the red dots are the best teaming strategies our approach returns when given the choice of only using the satellites in the benchmark team for the same random missions.

The mission assigned to our teams and the benchmark team is the same, and that is to observe at least 2 out of 4 parameters (MWIR, TIR, SAR, and Cloud Cover) every day for two weeks in a random location of Earth with high volcanic activity.

We observe that the best teaming strategies –in terms of the trade-off between probability of success and satellite usage– generated by our approach (the blue dots) dominate the teaming strategies generated from the benchmark team at NASA we described (the red dots). Therefore, our approach appears to create better teams and teaming strategies than the benchmark team that is already in use – according to our own metrics, which may be different from the ones used by

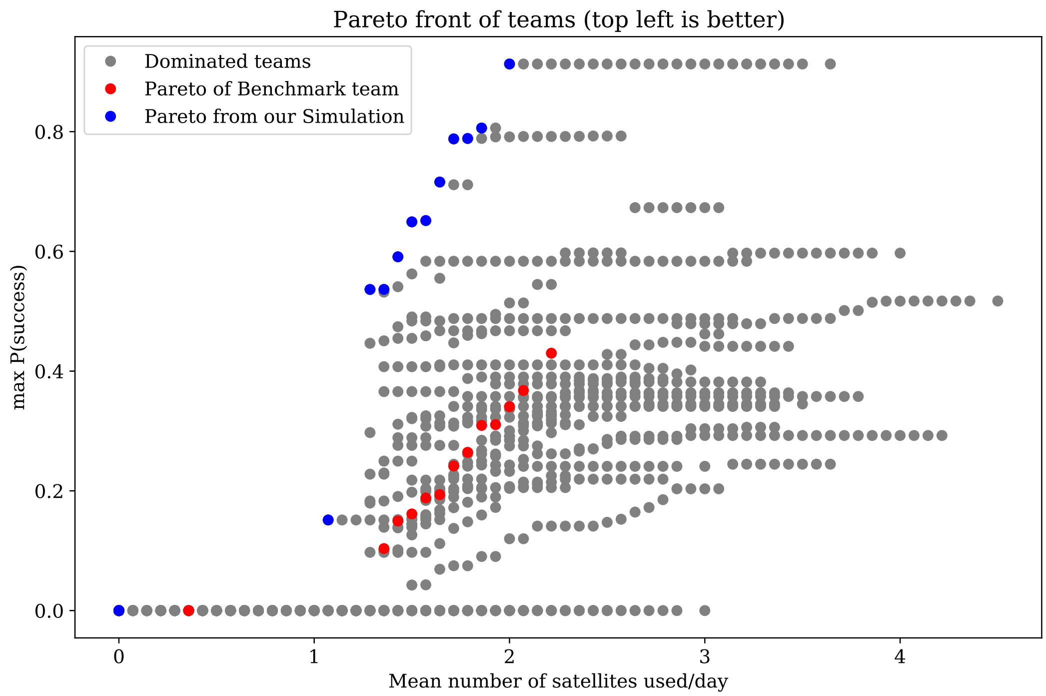


Fig. 3 Probability of mission success as a function of mean number of satellites used per day for various teaming strategies. Results demonstrate that our approach is capable of creating teams and daily assignments out of those teams that compete and even improve upon expert-designed teams.

NASA to make this decision.

Additionally, we provide some intermediate results showing the outputs of the different subsystems in this method.

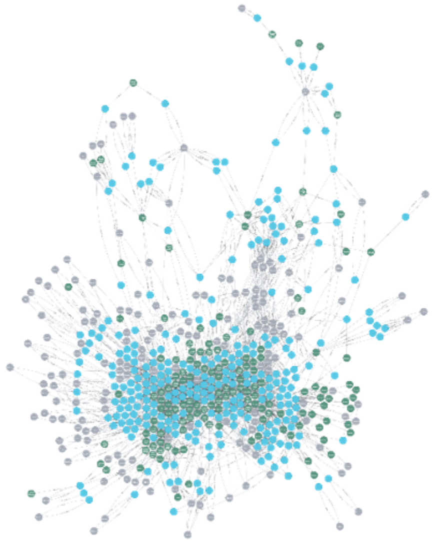
A. Results from Knowledge Representation

The main result of this subsystem is an initial Probabilistic Knowledge Base, which can be represented either as triples or as part of a Graph Database, as seen in Figure 4. The knowledge base has a total of 2460 nodes and 12500 relationships. Specifically, there are 635 satellites, 927 sensors, 168 Measurements, 519 Types of Sensors, and 169 Agencies, among others. The relationships are between sensors, satellites, measurements, events, with some examples being what types of sensors can measure what properties (with 3403 of those), and which measurements can be used to create higher level measurements and data products (a count of 519), as well as relations between events and their observable properties. For example, it finds that TIR Imaging multi-spectral radiometers can generally measure Earth's surface reflectance.

If we include the results from the knowledge extraction pipeline, there is an increase of nodes in the knowledge graph from 2460 to 3063 (an increase of 24.5%, with new measurements and properties leading the increase), while the number of relationships between nodes has increased from 12500 to 14500 (an increase of 16%, mostly in the field of what sensors can measure what).

B. Results from Reasoning over KG

The main result of this subsystem is a list of candidate satellites that can participate, i.e., that are potentially useful for the given mission. There are 161 active EO satellites orbiting Earth right now in the CEOS database. For the given mission "Active Volcano Monitoring", UniKER gives a total of 113 satellites among them as candidate satellites as far as sensor capabilities go. To validate our prediction, we compare our prediction with a handcrafted ground truth based on results from the CEOS database and the accesses of the satellite over the Kīlauea volcano. 89 satellites are identified



```
JPSS-1|VIIRS    isInstanceOf    VIIRS
JPSS-1  HOSTS   JPSS-1|ATMS
JPSS-1|ATMS isInstanceOf    ATMS
JPSS-1  HOSTS   JPSS-1|CERES
JPSS-1|CERES    isInstanceOf    CERES
JPSS-1  HOSTS   JPSS-1|CrIS
JPSS-1|CrIS isInstanceOf    CrIS
JPSS-1  HOSTS   JPSS-1|OMPS
JPSS-1|OMPS isInstanceOf    OMPS
MODIS   OBSERVES    Earth surface albedo
MODIS   OBSERVES    Cloud optical depth
HiRI    OBSERVES    Land surface imagery
CSG SAR OBSERVES    Snow cover
IVM OBSERVES    Ion Density, Drift Velocity, and Temperature
IASI    OBSERVES    Ozone profile
VIIRS   OBSERVES    Cloud top height
Vegetation  OBSERVES    Vegetation type
```

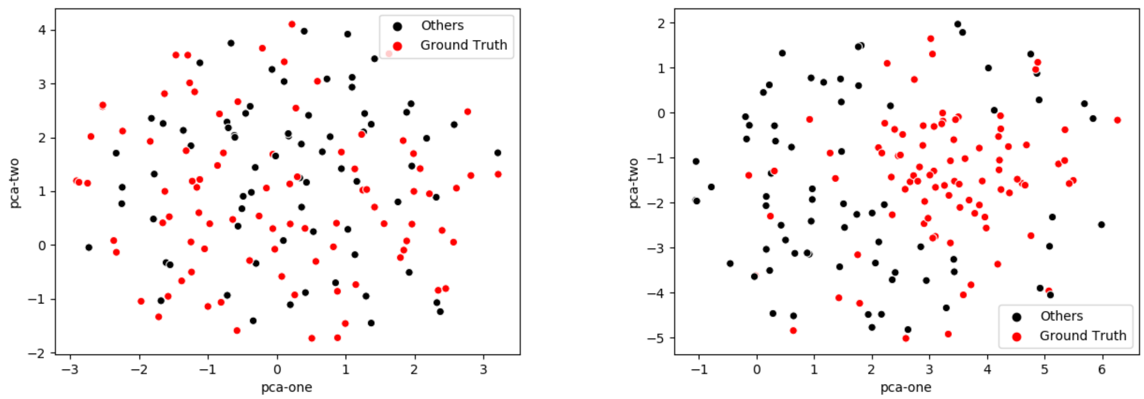
Fig. 4 Excerpts from the Graph and Triples databases, showing information on the satellites and sensors currently in orbit

in this ground truth as potentially useful satellites for our example mission from those in the CEOS database, according to their status as currently active, a path in the KG connecting their sensors to the observable properties (e.g. a TIR sensor can measure TIR Radiance - TIR radiance is generated by heat - a volcano eruption generates heat), as well as flying over the volcano at the right times. UniKER's logic prediction is able to find all 89 satellites given that all relationships linking measurements and events are present in the KG.

In addition, as our approach combines embedding and logical rules for better KG inference, in order to show the enhancement of logical rules over embeddings we compare our proposed framework with TransE. TransE is the most representative knowledge graph embedding model, which learns graph embeddings without the incorporation of logical rules. As shown in Figure 5, we have plotted the two principal components of the learned embeddings for all satellites. In particular, the red nodes represent the potential useful satellites given in our ground truth while the black nodes represent other satellites in the KG. We can observe that with only the embedding model, it is difficult to distinguish the useful satellites from the others, while with the help of logical rules, the quality of embeddings is improved dramatically.

C. Results from Sensor Planning Framework

The sensor planning framework outputs the probability of each participating satellite successfully detecting an event given the event happened (true positives), as well as the probabilities of a false positive, true negative, and false negative. To show an example of this intermediate result, we use the sensor planning algorithm to calculate the $p(Detected|EventHappens)$ for NASA's benchmark team in Table 1. We can see that, as we would expect from pieces of technology that usually cost millions of dollars, most sensors are good at what they do. This seems to prove that our system can accurately predict the probabilities we have mentioned.



(a) Result for TransE

(b) Result for Our Proposed Framework

Fig. 5 Visualization of a Principal Component Analysis of Embedding Vectors, showing that UniKER is capable of distinguishing useful satellites better than TransE.

Table 1 Benchmark Team From NASA and the computed probabilities of event detection from the sensors in each satellite

Satellites	Sensors	Observation	$p(Detected \vert EventHappens)$
Aqua	AIRS	Land surface temperature	0.99899652
	MODIS	Land surface temperature	0.99899652
		Fire temperature	0.99999966
	AMSU-A	Land surface temperature	0.99899652
		Cloud type	0.84130652
Terra	ASTER	Land surface topography	1.
		Land surface temperature	0.99899652
		Cloud type	0.99899652
	MODIS	Land surface temperature	0.99899652
		Fire temperature	0.99999966
Sentinel-1 A	C-Band SAR	Land surface topography	0.99899652
Sentinel-1 B	C-Band SAR	Land surface topography	0.99899652
GOES-15	Imager	Land surface temperature	0.99122158
		Fire temperature	0.99999882
GOES-17	ABI	Fire temperature	0.99999966
		Land surface temperature	0.99899652
		Cloud type	0.84130652

D. Results from Synthesis of Teaming Assignment

The synthesis subsystem outputs the probability of mission success and the optimal team given information from the Sensor Planning Framework and the KG. We generate a Pareto front of two objectives: minimizing the number of satellites and maximizing the max probability of mission success, where the mission is to monitor any eruptions of the Kīlauea volcano over a two-week time frame. To reduce the computation time, the process is parallelized to synthesize a Pareto front for each timestep, which is then aggregated into an overall plot (see Fig. 6). To cover the entire length of the mission, we generate a sample of all possible teaming combinations of the parallelized daily results, which are shown in gray. The red points represent the approximate Pareto front found from the sample.

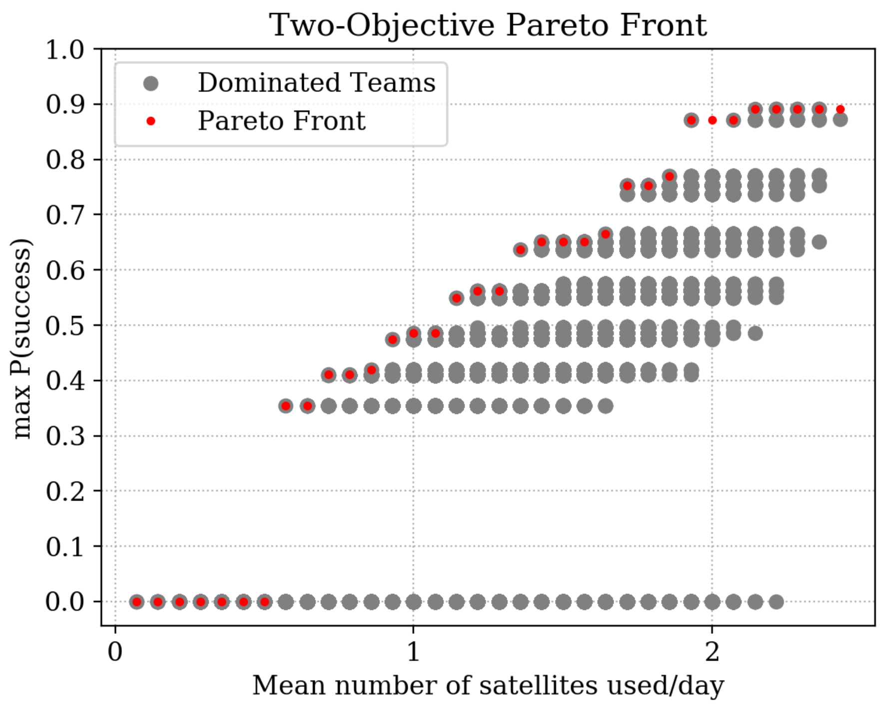


Fig. 6 Pareto front for a mission to monitor any eruptions of the Kīlauea volcano over two weeks, given a list of potential satellites. Each point on the plot corresponds to a different synthesized teaming plan.

The Pareto front provides a way to compare the benchmark team to the final team our system outputs. While the objective is to maximize probability, the other objective of the Pareto front can change depending on the priorities of the mission (e.g. minimizing energy usage instead of number of satellites, or some more complicated utility function).

IV. Conclusion

This paper described a new decentralized, context-based, on-board planning algorithm for future EO systems such as sensor webs, smart heterogeneous constellations, or federated systems. We explained in detail the three steps of our method, where each satellite first decides whether it can participate, then if it should participate, and finally a central node creates and formally verifies a good (potentially optimal) team for the mission. Two different teaming algorithms were described, based on the desire for more optimal centralized synthesis. Initial results for each step as well as the whole method were provided. These initial results show that our approach can create teams with a higher probability of success while using less satellites daily to perform a volcano monitoring mission than currently available systems that perform the same task.

The main limitation of this paper is the lack of more valid benchmarks and comparison systems to characterize the performance of the proposed approach. Although the comparison described in the Results section is a good first step, a more exhaustive performance benchmark is necessary. In order to further validate our system, we plan to create a benchmark platform with missions and benchmark teams against which our approach can be compared to the approaches mentioned in the introduction. We are particularly interested in comparing our system to other decentralized approaches such as the ACBBA-based approach described in [12], as well as the modified CBBA approach in [14].

A further limitation of the approach is that when it checks whether a satellite should participate in a mission, it does not check what that would imply for the current mission being performed by that satellite. We plan to extend the approach to a multi-mission framework in the future in order to address this limitation, as well as allow for different types of mission, such as multiple event detection, instead of the current single event detection approach. For example, we would like to test the system with a meta-mission to detect multiple natural disasters instead of just limiting it to a single one, like a Volcano Eruption. This requires expansions in all parts of our approach, among other things to support the testing of multiple simultaneous hypotheses.

In addition, in order to validate the a priori probabilities of success that are given as an output of our approach, we plan to enhance our current Monte-Carlo simulation system to include timelines for each of the physical observations for each random event and check whether the system's is capable of detecting the correct percentage of events (i.e. if the approach says a certain team will detect an event 90% of the time, it should detect 9/10 events).

Finally, we plan on further expanding the text mining capabilities to keep increasing the size of the KG, as well as incorporating more complex constraints on the mission specification that involve spacecraft resources, which are assumed as infinite and perfect right now. Besides, we plan on adding a capability for the system to understand and exploit sensor synergies. Furthermore, our team synthesis approach has two steps where a random algorithm is used due to the time complexity of the optimal approach. These will be improved upon by using algorithms currently available in the literature such as Tournament Selection and Genetic Algorithms.

Acknowledgements

This work has been funded by DARPA grant HR00112090027.

References

[1] Golkar, A., and Lluch, I., "The Federated Satellite Systems paradigm: Concept and business case evaluation," *Acta Astronautica*, Vol. 111, 2015, pp. 230 – 248. https://doi.org/https://doi.org/10.1016/j.actaastro.2015.02.009.

[2] Berger, M., Moreno, J., Johannessen, J. A., Levelt, P. F., and Hanssen, R. F., "ESA's sentinel missions in support of Earth system science," *Remote Sensing of Environment*, Vol. 120, 2012, pp. 84 – 90. https://doi.org/https://doi.org/10.1016/j.rse.2011.07.023, URL http://www.sciencedirect.com/science/article/pii/S003442571200065X, the Sentinel Missions - New Opportunities for Science.

[3] Boshuizen, C., Mason, J., Klupar, P., and Spanhake, S., "Results from the planet labs flock constellation," 2014.

[4] Vasquez, M., and Hao, J.-K., "A "Logic-Constrained" Knapsack Formulation and a Tabu Algorithm for the Daily Photograph Scheduling of an Earth Observation Satellite," *Computational Optimization and Applications*, Vol. 20, 1999. https://doi.org/10.1023/A:1011203002719.

[5] Wolfe, W., and Sorensen, S., "Three Scheduling Algorithms Applied to the Earth Observing Systems Domain," *Management Science*, Vol. 46, 2000, pp. 148–166. https://doi.org/10.1287/mnsc.46.1.148.15134.

[6] Gabrel, V., and Murat, C., *Mathematical Programming for Earth Observation Satellite Mission Planning*, chapter and pages, pp. 103–122. https://doi.org/10.1007/978-1-4757-3752-3_7, URL https://doi.org/10.1007/978-1-4757-3752-3_7.

[7] Di, L., Moe, K., and van Zyl, T. L., "Earth Observation Sensor Web: An Overview," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 3, No. 4, 2010, pp. 415–417. https://doi.org/10.1109/JSTARS.2010.2089575.

[8] Muylaert, J., Reinhard, R., Asma, C., Buchlin, J., Rambaud, P., and Vetrano, M., "QB50: an international network of 50 CubeSats for multi-point, in-situ measurements in the lower thermosphere and for re-entry research," *ESA Atmospheric Science Conference, Barcelona, Spain*, 2009, pp. 7–11.

[9] Fossa, C. E., Raines, R. A., Gunsch, G. H., and Temple, M. A., "An overview of the IRIDIUM (R) low Earth orbit (LEO) satellite system," *Proceedings of the IEEE 1998 National Aerospace and Electronics Conference. NAECON 1998. Celebrating 50 Years (Cat. No.98CH36185)*, 1998, pp. 152–159. https://doi.org/10.1109/NAECON.1998.710110.

[10] Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., Lee, R., Mandl, D., Frye, S., et al., "The EO-1 autonomous science agent," *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, Citeseer, 2004, pp. 420–427.

[11] Li, G., Xing, L., and Chen, Y., "A hybrid online scheduling mechanism with revision and progressive techniques for autonomous Earth observation satellite," *Acta Astronautica*, Vol. 140, 2017, pp. 308 – 321. https://doi.org/https://doi.org/10.1016/j.actaastro.2017.08.011, URL http://www.sciencedirect.com/science/article/pii/S009457651631373X.

[12] Li, G., "Online scheduling of distributed Earth observation satellite system under rigid communication constraints," *Advances in Space Research*, Vol. 65, No. 11, 2020, pp. 2475 – 2496. https://doi.org/https://doi.org/10.1016/j.asr.2020.02.018, URL http://www.sciencedirect.com/science/article/pii/S0273117720301101.

[13] Choi, H.-L., Brunet, L., and How, J. P., "Consensus-based decentralized auctions for robust task allocation," *IEEE transactions on robotics*, Vol. 25, No. 4, 2009, pp. 912–926.

[14] Gallud, X., and Selva, D., "Agent-based simulation framework and consensus algorithm for observing systems with adaptive modularity," *Systems Engineering*, Vol. 21, No. 5, 2018, pp. 432–454.

[15] Wright, R., "MODVOLC: 14 years of autonomous observations of effusive volcanism from space," *Geological Society, London, Special Publications*, Vol. 426, 2015. https://doi.org/10.1144/SP426.12.

[16] Chien, S. A., Davies, A. G., Doubleday, J., Tran, D. Q., Mclaren, D., Chi, W., and Maillard, A., "Automated Volcano Monitoring Using Multiple Space and Ground Sensors," *Journal of Aerospace Information Systems*, Vol. 17, No. 4, 2020, pp. 214–228. https://doi.org/10.2514/1.I010798, URL https://doi.org/10.2514/1.I010798.

[17] Selva, D., and Crawley, E. F., "VASSAR: Value assessment of system architectures using rules," *2013 IEEE Aerospace Conference*, 2013, pp. 1–21.

[18] Yadav, V., and Bethard, S., "A Survey on Recent Advances in Named Entity Recognition from Deep Learning models," , 2019.

[19] Smirnova, A., and Cudré-Mauroux, P., "Relation Extraction Using Distant Supervision: A Survey," *ACM Comput. Surv.*, Vol. 51, No. 5, 2018. https://doi.org/10.1145/3241741, URL https://doi.org/10.1145/3241741.

[20] Beltagy, I., Lo, K., and Cohan, A., "SciBERT: Pretrained Language Model for Scientific Text," *EMNLP*, 2019.

[21] Honnibal, M., and Montani, I., "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," , 2017. To appear.

[22] Bach, S. H., Broecheler, M., Huang, B., and Getoor, L., "Hinge-Loss Markov Random Fields and Probabilistic Soft Logic," *J. Mach. Learn. Res.*, Vol. 18, No. 1, 2017, p. 3846–3912.

[23] Mihalkova, L., and Mooney, R. J., "Bottom-up learning of Markov logic network structure," *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 625–632.

[24] Guo, S., Wang, Q., Wang, L., Wang, B., and Guo, L., "Knowledge Graph Embedding with Iterative Guidance from Soft Rules," , 2017.

[25] Cheng, K., Yang, Z., Zhang, M., and Sun, Y., "UniKER: A Unified Framework for Combining Embedding and Horn Rules for Knowledge Graph Inference," *Proceedings of the Graph Representation Learning and Beyond Workshop (ICML GRL+ 2020)*, 2020.

[26] Wang, Q., Wang, B., and Guo, L., "Knowledge base completion using embeddings and rules," *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[27] Rocktäschel, T., Singh, S., and Riedel, S., "Injecting logical background knowledge into embeddings for relation extraction," *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 1119–1129.

[28] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45.

[29] Yang, Y., and Gao, W., "An optimal adaptive Kalman filter," *Journal of Geodesy*, Vol. 80, No. 4, 2006, pp. 177–183.

[30] Chen, G., Wang, J., and Shieh, L. S., "Interval kalman filtering," *IEEE Transactions on Aerospace and electronic Systems*, Vol. 33, No. 1, 1997, pp. 250–259.

[31] Shuli Sun, "Multi-sensor optimal information fusion Kalman filter for discrete multichannel ARMA signals," *Proceedings of the 2003 IEEE International Symposium on Intelligent Control*, 2003, pp. 377–382.

[32] Chang, T.-K., and Mehta, A., "Control-Theoretical and Topological Analysis of Covariance Intersection Based Distributed Kalman Filter," *IEEE Control Systems Letters*, Vol. 2, No. 4, 2018, pp. 665–670. https://doi.org/10.1109/lcsys.2018.2844734, URL https://doi.org/10.1109%2Flcsys.2018.2844734.

[33] Boteanu, A., Howard, T., Arkin, J., and Kress-Gazit, H., "A model for verifiable grounding and execution of complex natural language instructions," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2649–2654.

[34] Mitsch, S., Ghorbal, K., Vogelbacher, D., and Platzer, A., "Formal verification of obstacle avoidance and navigation of ground robots," *The International Journal of Robotics Research*, Vol. 36, No. 12, 2017, pp. 1312–1340.

[35] Luckcuck, M., Farrell, M., Dennis, L. A., Dixon, C., and Fisher, M., "Formal Specification and Verification of Autonomous Robotic Systems: A Survey," *ACM Comput. Surv.*, Vol. 52, No. 5, 2019.

[36] Kress-Gazit, H., Lahijanian, M., and Raman, V., "Synthesis for Robots: Guarantees and Feedback for Robot Behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 1, No. 1, 2018, pp. 211–236. https://doi.org/10.1146/annurev-control-060117-104838, URL https://doi.org/10.1146/annurev-control-060117-104838.

[37] Clarke, E., Grumberg, O., and Peled, D., *Model Checking*, MIT Press: Cambridge, Massachusetts, 1999.

[38] Hansson, H., and Jonsson, B., "A Logic for Reasoning about Time and Reliability," *Formal Aspects of Computing*, Vol. 6, 1995.

[39] Kwiatkowska, M., Norman, G., and Parker, D., "PRISM 4.0: Verification of Probabilistic Real-Time Systems," *Computer Aided Verification*, edited by G. Gopalakrishnan and S. Qadeer, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 585–591.